НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки
(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління
(назва кафедри)

"На правах рукопису"
УДК *004.056.5*

«До захисту допущено»
Завідувач кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ ” 20 *18* р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю *126 Інформаційні системи та технології*
(код та назва спеціальності)

ОПП *Інформаційні управляючі системи та технології*
(код та назва спеціалізації)

на тему: *Інформаційна система захисту персональної інформації на основі комплексного підходу до шифрування та збереження даних*

Виконав: студент *VI* курсу *ІС-з71мп*
(шифр групи)

Скидан Дмитро Олександрович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *доц., к.т.н., доц. Жданова О.Г.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант *д.т.н., проф. Томашевський В.М.* _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018

РЕФЕРАТ

Магістерська дисертація: 104 с., 1 додаток, 26 рисунків, 29 таблиць, 23 джерела.

Актуальність. У зв’язку з останніми крадіжками персональних даних у facebook, надання інформації спец службам про користувачів в системах VK, Viber, Telegram, користувачам не залишається можливостей захистити свої дані від зламу та крадіжки. На жаль, на даний момент, немає жодної системи обміну та зберігання персональних даних, яка 100% захистить ваші дані. Тому ринок на даному етапі потребує такої системи.

Мета дослідження – підвищення ступеню захищеності персональних даних, та надання користувачу повного контролю над ними.

Для досягнення мети необхідно виконати наступні **завдання**:

- розробити функціональну можливість локального збереження особистих даних на кінцевих пристроях користувача;
- розробити можливість передачі даних на інші пристрої, у зв’язку з обмеженими ресурсами пам’яті, або для зберігання резервних копій даних;
- реалізувати можливість обміну персональними даними між користувачами у вигляді повідомлень файлів та медіа;
- розробити комплексний асиметричний алгоритм для захисту каналу передачі даних;
- розробити функціональну можливість контролю даних на всіх пристроях користувачем;
- спроектувати та розробити програмне забезпечення клієнтів у вигляді мобільного застосунку.

Об’єкт дослідження – процес збереження або обміну персональних даних користувача.

Предмет дослідження – інформаційна система захисту персональної інформації на основі комплексного підходу до шифрування та збереження даних.

Методи дослідження, застосовані у даній роботі, базуються на розробці власного комплексного алгоритму захисту даних. Використання асинхронних,

синхронних алгоритмів захисту інформації. Розробка комплексного алгоритму на основі гібридного підходу до шифрування.

Наукова новизна одержаних результатів полягає у впровадженні нового підходу до збереження даних; впровадженні нових способів передачі даних; розробці нових комплексних алгоритмів захисту даних в незахищеному каналі;

Публікації:

Тези: Комплексний асиметричний алгоритм шифрування з динамічним ключем. Автори: Скидан Д.О, Жданова О. Г. Конференція: Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2018) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 29-30 грудня 2018 р.

Стаття: Аналіз симетричних алгоритмів шифрування для впровадження у гібридну криптосистему. Автор: Скидан Д. О. Науковий журнал: Актуальні наукові дослідження в сучасному світі – iSience – 2018.

ЗАХИСТ, ПЕРСОНАЛЬНІ ДАНІ, ПЕРЕДАЧА ДАНИХ, ЗАХИЩЕНІ ПОВІДОМЛЕННЯ, КОНТРОЛЬ ПЕРСОНАЛЬНИХ ДАНИХ, ЗБЕРЕЖЕННЯ ДАНИХ, ЛОКАЛЬНИЙ КОНТРОЛЬ ДАНИХ, СИСТЕМА КЕРУВАННЯ І РОЗПОВСЮДЖЕННЯ ПЕРСОНАЛЬНИХ ДАНИХ

ABSTRACT

Master's dissertation: 106 p., 1 appendix, 26 figures, 29 tables, 23 sources.

Topicality. Due to the latest theft of personal data in facebook, providing information to special services about users from VK, Viber, Telegram, users do not have the ability to protect their data from hack and theft. Unfortunately, at the moment, there is no system for exchanging and storing personal data that will protect your data on 100%. Therefore, the market at this stage needs such a system. Therefore, the market needs such a system, at this step.

Aim of research - increasing the level of personal data protection and providing full data control to the user.

To achieve the aim, should accomplish the following tasks:

- develop a functional possibility of personal data local storage of on the user devices;
- develop the transfer data ability to other devices, due to limited memory resources, or to back up data;
- realize the possibility of exchanging personal data between users, such as message files and media;
- develop a comprehensive asymmetric algorithm for protection the data channel;
- develop a functional possibility of data control on all devices by the user;
- design and develop client software in the form of a mobile application.

Object of the research – the process of storing or exchanging user personal data.

Subject of research - Personal information protection system based on complex approach of Data Encryption and Storage.

The used **Research Methods** are based on the development of a self-contained complex data protection algorithm. Use of asynchronous, synchronous data protection algorithms. Development of complex algorithm based on hybrid encryption approach.

The scientific novelty of the obtained results is the introduction of a new approach to data storage; introduction of new methods of data transferring; development new complex algorithms for data protection in an unprotected channel;

Publications:

Theses: The complex asymmetric encryption algorithm with a dynamic key. Authors: Skydan D., Zhdanova O. Conference: “Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2018)” – Kyiv: The National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" 29-30 Dec. 2018.

Article: Analysis of symmetric encryption algorithms for implementation in the hybrid cryptosystem. Authors: Skydan D. Icientific journal: “Актуальні наукові дослідження в сучасному світі” – iSience – 2018.

PROTECTION, PERSONAL DATA, DATA TRANSMISSION, PROTECTED MESSAGES, PERSONAL DATA CONTROL, DATA PROTECTION, LOCAL DATA CONTROL, PERSONAL DATA CONTROL AND DISTRIBUTION SYSTEM

ЗМІСТ

ВСТУП

Сьогодні дані формують центральну роль у функціонуванні наших суспільств та економік. Це означає, що вони є одним із найцікавіших ресурсів XXI століття: "дані є електрикою нашої нової економіки", - заявляє інформатор Cambridge Analytics. Зі збільшенням кількості соціальних мереж, онлайн магазинів, месенджерів, дані все більше і більше зацікавлюють хакерів.

На даних момент, майже всі особисті дані - зацифровані. Крадіжка даних, це сфера, що викликає велике занепокоєння серед споживачів протягом багатьох років. З кожним роком, особистих даних в мережі стає все більше, а крадіжки стають все гучнішими.

Корпорації які володіють великими базами даних про користувачів – цікавлять хакерів все більше. Деякі корпорації стають на спокусі продажу даних іншім особам. Тільки за останні роки, були спалахи скандалів про корпорації VK, Viber, Whatsapp і найбільший скандал з крадіжки даних пов’язаний з facebook.

В людей більше нема довіри до сервісів використання персональної інформації. За останні роки, спостерігається спад інтересу та попиту до систем обміну та зберігання персональної інформації.

Система розглянута в дисертації пропонує новий підхід до зберігання і обміну інформацією. Основна ідея полягає в збереженні та контролі інформації на стороні користувача. Користувач має абсолютний контроль в обміні та керуванні своїми даними. Сервера, всього-на-всього, виступають мостом в трансфері даних, але не зберігає їх.

Система надає альтернативні способи збереженні даних. Дані можуть бути збереженні як на активних пристроях – ноутбуки чи телефони, а також на окремих пристроях-сховищах, які можуть бути розташовані де завгодно і підключені коли завгодно.

Система може виконувати функцію підсистеми в бізнес рішеннях. Одна з цілей розповсюдження, це впровадження в корпораційні системи. Компанії, або цілі корпорації, мають змогу комунікувати з користувачами своєї системи без позбавлення контролю над даними, але захищаючи, зі своєї сторони, оболонку

комунікації.

Система має мету кардинальної зміни ставлення до обробки персональних даних, та поверненні довіри користувачів до систем передачі і збереженні персональних даних.

1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗАХИСТУ ПЕРСОНАЛЬНОЇ ІНФОРМАЦІЇ НА ОСНОВІ КОМПЛЕКСНОГО ПІДХОДУ ДО ШИФРУВАННЯ ТА ЗБЕРЕЖЕННЯ ДАНИХ

1.1 Опис бізнес – процесів

1.1.1 Опис процесу діяльності

Сервіс надає змогу користувачу або бізнесу комплексний захист даних. Особливість цього сервісу в забезпеченні захисту персональних даних у вигляді повідомлень між користувачами, аудіо та відео дзвінків, медіа-файлів або документів. Основна ідея сервісу в контролі і збережені даних на стороні користувача. Сервер в цій системі виступає лише як трансфер між пристроями.

Кінцевими підсистемами для користувача або бізнесу виступають застосунки на платформах (iOS, macOS, Android, Windows) або веб-сайт. Всі дані зберігаються локально на мобільних пристроях/персональних комп’ютерах. Також додається фізичний пристрій-сховище. Який встановлюється за розсудом користувача та після під’єднання до мережі інтернету з’являється можливість відвантаження даних з кінцевих пристроїв на пристрій-сховище. Для впровадження в бізнес є можливість придбання та встановлення власного трансфер сервісу. В такому вигляді система може працювати повністю локально, за єдиним запитом перевірки ліцензії.

Захист даних формується за використанням симетричних та асиметричних алгоритмів шифрування. Особисті дані які пересилаються під час трансферу на локальний пристрій-сховище або у вигляді повідомлень з чатів захищаються більш ретельно з комплексним підходом шифрування. Комплексний підхід базується на використанні змішаної форми шифрування. До вже відомих типів шифрування додається новий тип - з змінним ключем. Ключ залежить від попередніх блоків даних.

Ефективність такого типу шифрування полягає в наступному:

- постійна зміна ключа на новій ітерації даних;
- немає чіткого зберігання ключа, він зберігається в історії повідомлень за умовою отримання ключа з попередньої ітерації;

- наявність механізму поширення помилки: якщо при передачі відбудеться зміна одного біта шифротексту, дана помилка поширюється і на наступний блок. однак на наступні блоки (через один) похибка не поширюється;
- зловмисник має можливість додати блоки до кінця зашифрованого повідомлення, доповнюючи тим самим відкритий текст (проте без ключа виходить сміття. а з використанням hash функції для підтвердження цілісності повідомлення, зводить цей тип атаки нанівець).

Підсервіс у вигляді мобільного застосунку буде представлений на платформі iOS. Підсервіс надає можливість передачі повідомлень між користувачами, збережені аудіо та відео дзвінків, медіа-файлів або документів.

В мобільному застосунку присутній захищений браузер, який дозволяє користувачам переглядати веб-сайти без будь-якого вистежування їхньої діяльності, або особи.

Для обміну повідомленнями був обраний протокол XMPP (Extensible Messaging and Presence Protocol - протокол розширюваних повідомлень та присутності(укр.)) - це протокол зв'язку для проміжного програмного забезпечення, орієнтованого на повідомлення, на основі XML (Extensible Markup Language). Це дає змогу обмінюватися структурованими, але розширюваними даними між об'єктами двох або більше мереж у режимі реального часу. Спочатку був названий Jabber. Протокол розроблювався як легко розширювальний, тому він часто використовується для VoIP дзвінків, передачі файлів, обмін аудіо та відео даними, у використанні інтернет речей (IoT), ігор та багато де ще [1].

Файлова система та сховище також шифруються в залежності від типу даних. В підсистемі - застосунку використовується база даних, де зберігаються історія повідомлень, а також медіа файли та документи. Шифрування сховища та бази даних основана в більшості від вхідних даних у вигляді пароля користувача.

1.1.2 Актори і функції

Функціональна модель всієї системи базується на виконанні таких процесів:

- забезпечення трансферу даних з локальної системи пристроїв до пристроїв-сховищ;
- забезпечення відвантаження та завантаження даних з/на пристроїв-сховищ;
- забезпечення комплексного шифрування даних на різних сховищах;
- забезпечення комплексного шифрування даних під час трансферу або відправці повідомлення;
- надання послуг локального трансферу бізнесу.

Функціональна модель підсистеми у вигляді мобільного застосунку:

- створення персонального аккаунту;
- збереження та перегляд контактів (інших користувачів системи) ;
- можливість перегляду статусу користувачів;
- створення захищеного чату;
- збереження історії повідомлень;
- створення захищеного аудіо дзвінка;
- створення захищеного відео дзвінка;
- захищене збереження медіа даних;
- захищене збереження файлів;
- можливість відвантаження резервної копії на пристрій-сховище;
- можливість перегляду захищеного браузеру.
- можливість захистити вхід в застосунок після неактивного стану пін-кодом

Одною з основних задач було реалізувати комплексне шифрування для різних типів даних. Так як основна ідея базується в контролі даних і збереженні їх на стороні користувача, основною вразливою точкою в захисті системи є трансфер даних або відправка повідомлень. Тут пакети даних захищаються симетричним шифруванням, асиметричним шифруванням та динамічним ключем. Такий комплексний підхід зменшує відсоток вразливості даних.

Актором в системі виступає користувач. Кінцевий в підсистемі – мобільний застосунок, дає можливість додавати інших користувачів до списку контактів, створювати чати та дзвінки, користуватись браузером, зберігати файли та створювати резервну копію даних. Діаграма варіантів використання наведена на рисунку 1.1. У таблиці 1.1 наведено опис визначених варіантів використання.

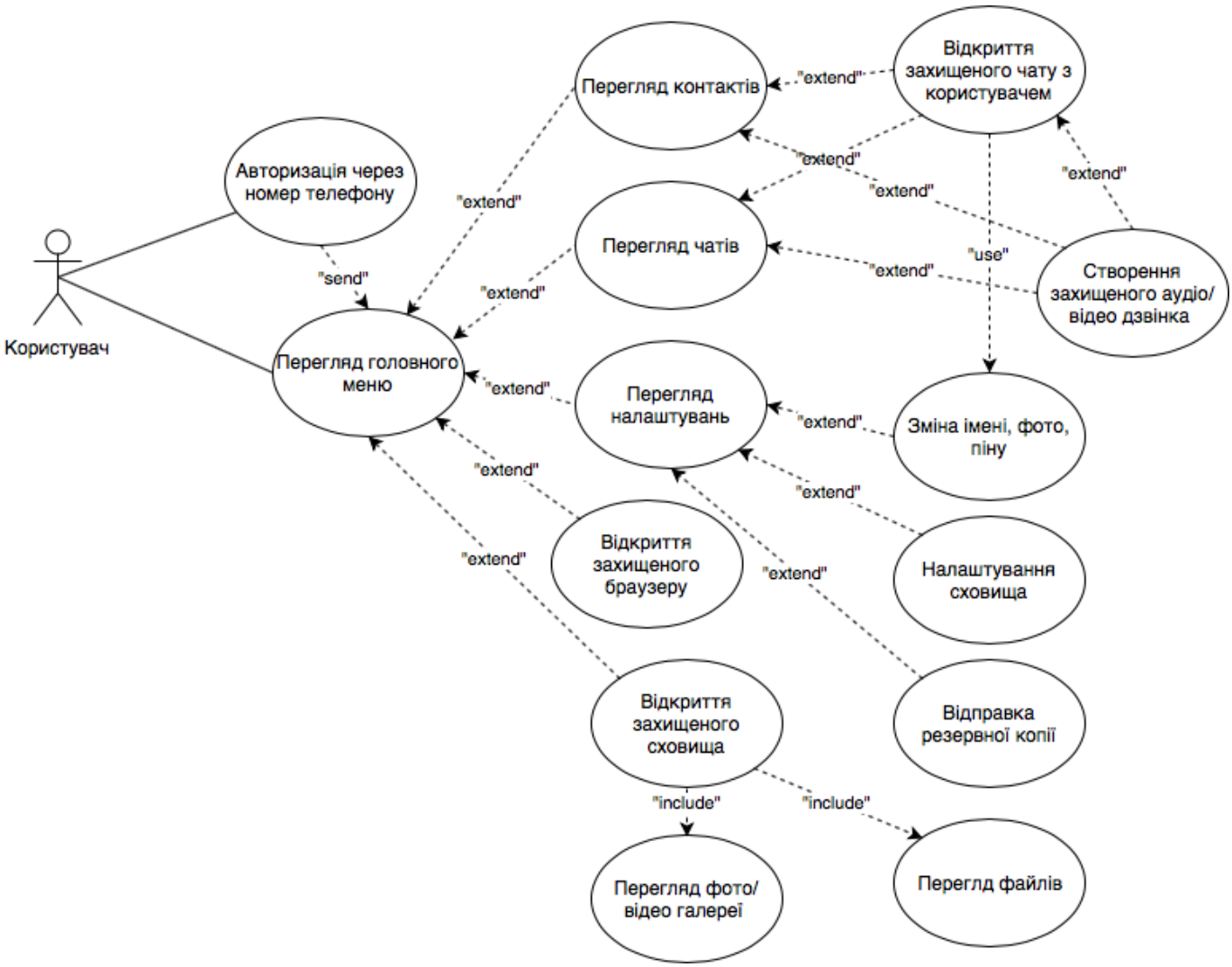


Рисунок 1.1 – Діаграма варіантів використання підсистеми – мобільний застосунок

Таблиця 1.1 – Опис варіантів використання

Актор	*Назва варіанту*	*Опис*	*Пріоритет*
Користувач	Авторизація через номер телефону	Користувач авторизується для подальшого користування сервісом через телефон, підтверджуючи номер смс-повідомленням	Високий
	Перегляд меню	Користувач має можливість переглядати пункти меню, після авторизації	Високий
	Перегляд контактів	Користувач має можливість переглядати список та додавати інших користувачів.	Високий
	Перегляд чатів	Користувач має можливість переглядати список чатів, відкривати їх.	Високий
	Створення захищеного відео/аудіо дзвінка з користувачем	Користувач з іншим користувачем має можливість створити захищений відео/аудіо дзвінок з.	Середній
	Створення захищеного чату з користувачем	Користувач з іншим користувачем може створити захищений чат для обміну повідомленнями.	Високий

Актор	*Назва варіанту*	*Опис*	*Пріоритет*
	Перегляд налаштувань	Користувач може переглядати свої налаштування	Високий
	Зміна фото, імені, піну	Користувач має можливість змінити пін, фото або ім'я в налаштуваннях.	Низький
	Налаштування сховища	Користувач має можливість налаштувати пароль від сховище та обмежити доступ	Високий
	Відправка резервної копії	Користувач має можливість відправити резервну копію на пристрій-сховище.	Високий
	Відкриття захищеного браузеру	Користувач має можливість відкрити захищений браузер без вистежування діяльності або особи	Середній
	Відкриття захищеного сховища	Користувач має можливість відкрити захищене сховище та керувати даними	Високий
	Перегляд фото/відео галереї	Користувач має можливість переглядати медіа файли та керувати ними	Високий
	Перегляд файлів	Користувач має можливість переглядати файли та керувати ними	Високий

1.1.3 Структура бізнес процесів

Сервіс має декілька процесів діяльності взаємодії з пристроєм-сховищем, з медіа файлами, повідомленнями та ін. Процес відправки повідомлення користувачу описаний в діаграмі діяльності у додатку А.

Процес авторизації представлений на діаграмі діяльності рисунок 1.2.

В таблиці 1.2 представлена залежність функціональних вимог від кроків бізнес-процесів.

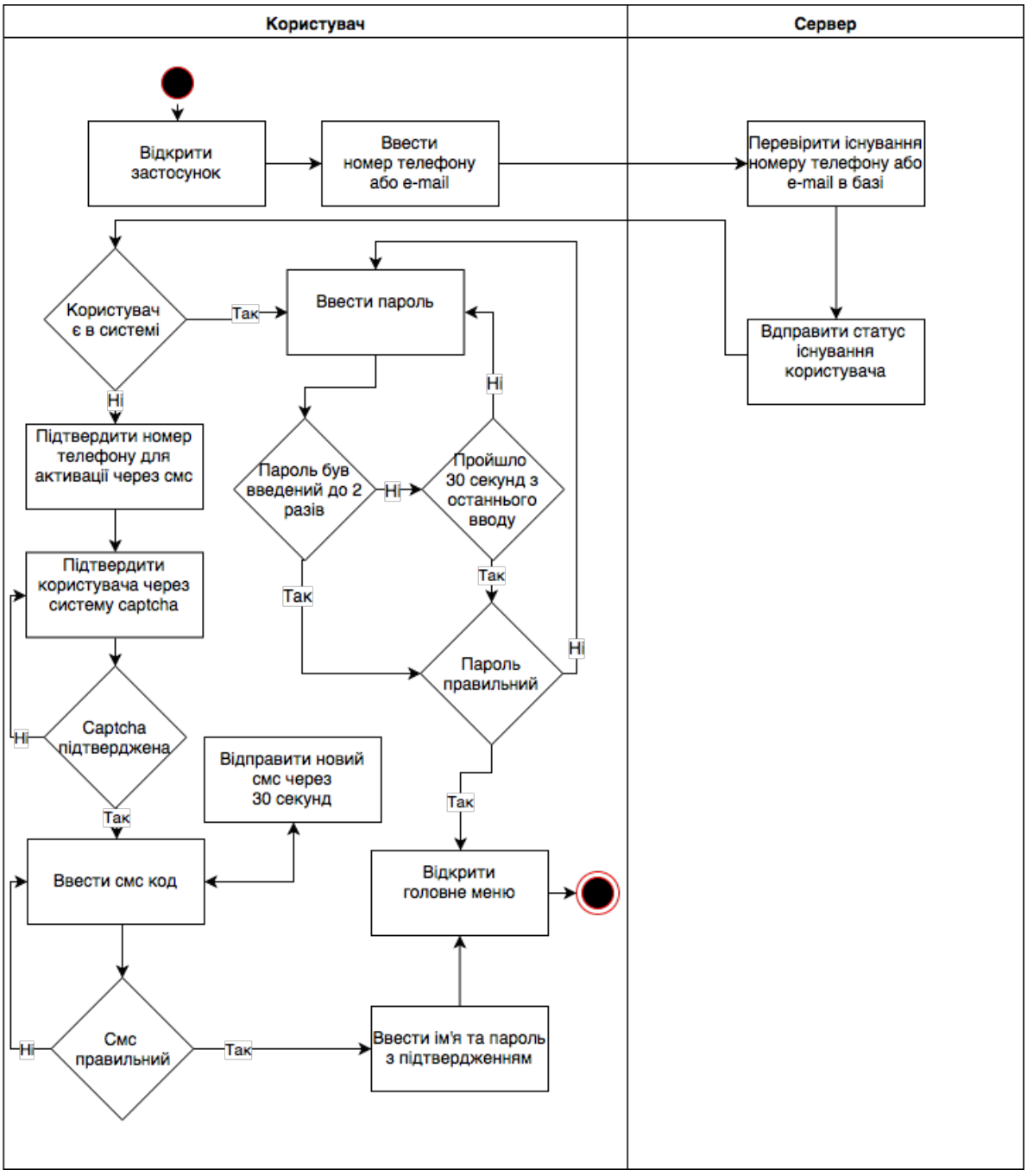


Рисунок 1.2 – Діаграма діяльності авторизації в системі

Таблиця 1.2 – Залежність функціональних вимог від кроків бізнес процесів

№	*Крок бізнес-процесу*	*Функціональна вимога*
1.	Перегляд списку користувачів	1. Сервіс надає можливість відображення списку локально збережених користувачів 1.1 Можна побачити статус кожного користувача в сервісі 1.2 Можна відправити повідомлення користувачу або здійснити дзвінок
2.	Авторизація через мобільний номер	2. Сервіс надає форму для авторизації клієнта за допомогою мобільного телефону. 2.1 Якщо клієнт не зареєстрований, він підтверджує свій номер за допомогою SMS та вводить ім'я і пароль. 2.1.1 Сервіс проводить авторизацію. 2.2 Якщо клієнт зареєстрований, він вводить пароль і входить в систему
3.	Перегляд налаштувань	3. Сервіс надає можливість змінити налаштування профілю 3.1 Користувач отримує можливість змінити фото. 3.2 Користувач отримує можливість змінити ім’я. 3.3 Користувач отримує можливість змінити пін. 3.4 Користувач отримує можливість налаштувати сховище

№	*Крок бізнес-процесу*	*Функціональна вимога*
		3.5 Користувач отримує можливість відправити резервну копію
4.	Використання захищеного браузеру	4. Сервіс надає можливість використовувати захищений браузер без вистежування діяльності та особистості
5.	Чат з користувачем	5. Сервіс надає можливість користування захищеним чатом з обраним користувачем. 5.1 Користувач може вводити текст через клавіатуру в телефоні. 5.2 Користувач може використовувати створені ним емоції 5.3 Користувач може відправляти файли або медіа.
6.	Дзвінок з користувачем	6. Сервіс надає можливість створення захищеного дзвінка. 6.1 Користувач може вмикати та вимикати камеру. 6.2 Користувач може вмикати та вимикати мікрофон.
6.	Додавання друга	6. Сервіс надає можливість додати друга за унікальним ID, або автоматично з тел. книги.
7.	Перегляд сховища	7. Сервіс надає можливість відображення файлів локального сховища. 7.1 Можна переглядати медіа 7.2 Можна переглядати файли

1.2 Опис постановки задачі

1.2.1 Призначення розробки

Система призначена для централізованого захищеного збереження та передачі особистих даних. Передача даних у вигляді повідомлень або трансферу на локальні носії, або збереження даних захищені комплексним підходом шифрування. Керування та збереження даних на стороні користувача досягається локальним збереження на кінцевих пристроях або на пристроях-сховищах.

1.2.2 Цілі та задачі розробки

Основною метою розробки є забезпечення надійного захисту персональних даних, швидкого шифрування та дешифрування даних, забезпечення надійного каналу передачі даних.

Для впровадження в бізнес, основною метою є забезпечення повного контролю даних бізнесом в особистій мережі.

Основні цілі досягаються за рахунок досягнення локальних:

- пришвидшення відправки та отримання захищених повідомлень в чаті за допомогою технології Socket;
- оптимізація комплексного алгоритму шифрування блоків в передачі даних шляхом знаходження слабких місць в алгоритмі;
- забезпечення високошвидкісної та надійної передачі даних при трансфері за рахунок поділу даних на блоки, та шифрування кожного блоку в захищеному потоці;
- забезпечення надійної системи локального збереження даних за рахунок комплексного підбору алгоритму шифрування;
- надання незалежного серверу який виступає як трансфер даних, а не накопичувач;
- надання для бізнесу незалежно-копійовану трансферну систему для впровадження її як підсистему в інші процеси.

Для досягнення цих цілей необхідно виконувати такі завдання:

- розробити можливість передачі даних на інші пристрої, у зв’язку з обмеженими ресурсами пам’яті, або для зберігання резервних копій даних.

- розробити комплексний асиметричний алгоритм для захисту socket каналу передачі даних;
- реалізувати можливість обміну персональними даними між користувачами у вигляді повідомлень та медіа-файлів;
- розробити метод передачі даних без буферного збереження пакетів на стороні серверу;
- розробити функціональну-можливість контролю даних на всіх пристроях користувачем;
- спроектувати програмне забезпечення для пристрою-сховища, який має виконувати роль сховища даних, як для буферу при передачі особистих даних, так і для додаткового зберігання даних або резервних копій;
- розробити сервіс обміну та передачі даних для взаємодії пристроїв користувача, а також спроектувати можливість впровадження сервісу як підсистему у бізнес системи;
- спроектувати та розробити програмне забезпечення клієнтів на прикладі мобільного застосунку;
- спроектувати функціональну можливість розподіленого локального збереження особистих даних на кінцевих пристроях користувача;

1.3 Рішення з інформаційного забезпечення

1.3.1 Вхідні дані

Вхідні дані поділяються на три основних види: основна інформація про користувача, трансльовані дані модуля текстових повідомлень та модуля дзвінків, персональні накопичувані дані.

Основна інформація про користувача отримується ручним вводом:

- номер телефону;
- ім’я користувача;
- пароль;
- фото;
- пін код;

Трансльовані дані модуля текстових повідомлень та модуля дзвінків:

- первісне повідомлення від користувача;
- потік аудіо від користувача;
- потік відео від користувача;

Персональні накопичувані дані:

- медіа файли;
- текстові файли;
- архіви;
- резервна копія;

1.3.2 Вихідні дані

Вихідними даними в сервісі виступають розшифровані дані та інформація про користувачів:

- статус користувача;
- отримані розшифровані повідомлення;
- отримані розшифровані файли;
- отримані розшифровані аудіо та відео потоки.

Сервіс складається з сервера обміну та передачі даних, пристрою-сховища, кінцевих клієнтів (iOS, macOS, Android, Windows).

При розгляданні підсистеми мобільного застосунку маємо досить велику SQL базу даних, яка представлена у додатку А.

Сервер представлений на базі протоколу XMPP, та має NoSQL базу. База дуже велика, тому для зручності представимо поля та сутності які, насамперед, використовуємо.

Так як основний принцип побудови сервісу базується на тільки трансферному значенні, то ніякого зберігання даних на сервері нема. Із даних, які необхідно зберігати – це користувацькі дані. Користувацькі дані зберігаються для уникнення постійного вводу персональних даних при вході в застосунок. Вони використовуються для створення чат-каналів, аудіо чи відео, та для завантаження контактів, трансфері даних. Всі користувацькі дані отримуються на перших декількох кроків при авторизації та більше не турбують користувача.

Представимо сутність для зберігання зареєстрованих користувачів. ER-модель сутності користувача представлена на рисунку 1.4 та складається з таких атрибутів:

- name (ім’я);
- phoneNum (номер телефону);
- id (унікальний ідентифікатор в системі);
- photo (аватар);
- relatedUsers (масив друзів).

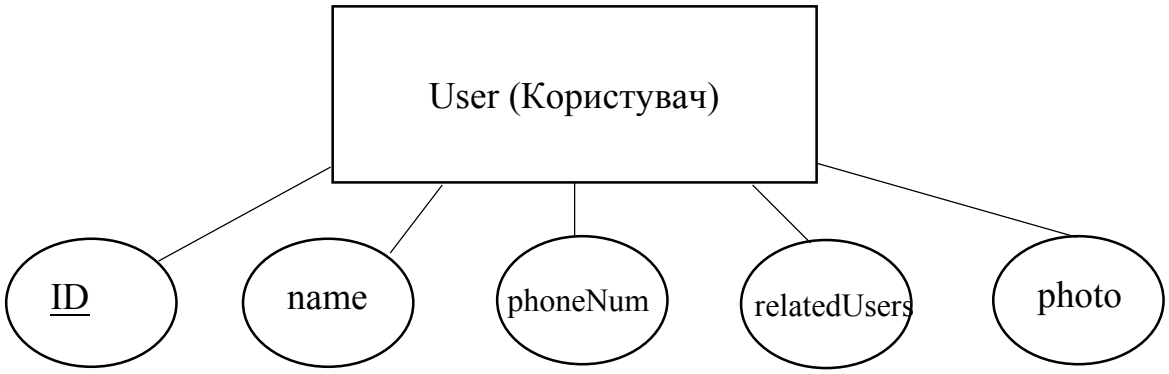


Рисунок 1.4 - ER-модель Користувача

Користувачу було б зручно зберігати історію своєї переписки та дзвінків. Але основною ідеєю сервісу є зберігання інформації на стороні користувача, тому історія досягається шляхом зберігання повідомлення на стороні відправника, або на пристрої-сховищі. Система є багатопоточна, тому паралельно виконує процеси відправки повідомлення, файлів, отримання їх, синхронізації і збереження. Що стосується зберігання аудіо потоків, тут ситуація складніша. По перше, потік є неперервним і має великий розмір. Також, окрім корисної інформації, потік має і не корисну (шуми, постійні звуки). З цього випливає, що зберігання дзвінків є не доречним, тому будемо зберігати тільки переписки, а дзвінки ні.

Модель повідомлень має структуру, яка описана на рисунку 1.5. Модель має такий список атрибутів:

- deliveryStatus (статус повідомлення);
- receiverID (отримувач повідомлення);
- senderID (відправник повідомлення);

- message (перекладений зміст повідомлення).
- date (дата відправки)

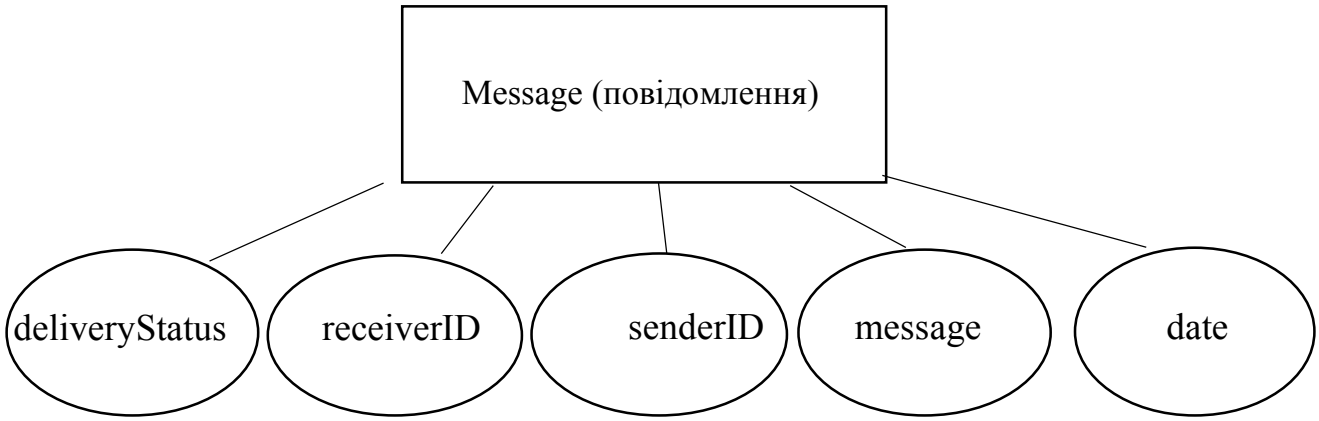


Рисунок 1.5 – Структура повідомлення

1.3.3 Структура масивів інформації

Відправка повідомлень здійснюється через XMPP протокол. Повідомлення основані на XML структурі – рисунки 1.6, 1.7.

	initiating to receiving	receiving to initiating
to	JID of receiver	JID of initiator
from	JID of initiator	JID of receiver
id	ignored	stream identifier
xml:lang	default language	default language
version	XMPP 1.0+ supported	XMPP 1.0+ supported

Рисунок 1.6 - Кореневі атрибути потоку повідомлень

```
<stream:stream
    from='juliet@im.example.com'
    to='im.example.com'
    version='1.0'
    xml:lang='en'
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'>
  <message>
    <body>foo</body>
  </message>
</stream:stream>
```

Рисунок 1.7 - Приклад повідомлення

Додатковими атрибутами є:

xmlns='jabber:client'– простір імен

xmlns:stream='http://etherx.jabber.org/streams –потік

<message xmlns='jabber:client'><body> - повідомлення

Висновки до розділу

В даному розділі було описано предметне середовище та пояснені бізнес процеси системи. Була обґрунтована доцільність створення системи для захисту персональних даних.

Розглянуто процес діяльності, варіанти використання сервісу, описана роль користувача сервісу, сформульовані призначення та цілі створення сервісу.

Описана функціональна модель, чітко вказані задачі по яким формувалась функціональна модель.

Був проведений аналіз вхідних та вихідних даних, описана структура бази даних та масивів інформації. Для формату передачі повідомлень був вибраний XML формат – це один з двох найзручніших та популярних форматів, поруч з JSON. Другий більше підходить для передачі даних в REST запитах.

2 МОДЕЛІ ТА МЕТОДИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗАХИСТУ ПЕРСОНАЛЬНОЇ ІНФОРМАЦІЇ НА ОСНОВІ КОМПЛЕКСНОГО ПІДХОДУ ДО ШИФРУВАННЯ ТА ЗБЕРЕЖЕННЯ ДАНИХ

2.1 Змістовна постановка задачі

З кожним днем зростає кількість крадіжок та інцидентів, пов'язаних з даними. Кібератаки продовжують розвиватися, хакерами знаходяться нові витончені методи враження даних інтернет-користувачів. Щоб протидіяти таким спробам захист інформації став невіддільною мірою у сучасному світі кібербезпеки. Шифрування - це один зі способів захисту дискретної інформації, що передається в Інтернеті.

На сьогодні створено багато складних алгоритмів для шифрування чутливої до ураження інформації (тобто переводу її в незрозумілий формат). Оскільки найбільш вразливою частиною системи є передача даних, то однією з основних задач є знаходження комплексного підходу до захисту каналів передачі.

Зашифровані даних можуть бути розшифровані лише за допомогою належних функцій, відомих як "криптографічні ключі". В основному криптографічний ключ - це пароль, який використовується для шифрування та розшифрування інформації.
Є два типи криптографічних ключів, відомі як симетричні (секретні) та асиметричні, вони відповідають алгоритмам шифрування, а саме симетричному та асиметричному. Симетричне шифрування - це звичайний метод шифрування. Симетричне шифрування виконується за допомогою лише одного секретного ключа, відомого як "Симетричний ключ", яким володіють обидві сторони. Цей ключ застосовується для кодування та декодування інформації. Відправник використовує цю клавішу перед надсиланням повідомлення, а приймач використовує його для розшифрування кодованого повідомлення [2].

Це досить простий спосіб, і, як наслідок, це займає не багато часу. Коли справа доходить до передачі величезних даних, симетричні ключі є кращими. Цезарний шифр - це хороший приклад симетричного шифрування. Сучасні підходи симетричного шифрування виконуються з використанням таких алгоритмів, як RC4, AES, DES, 3DES, QUAD, Blowfish та ін [3].

Асиметричне шифрування - це відносно новий і складний режим шифрування. Він комплексний, оскільки він включає в себе два криптографічних ключі для забезпечення безпеки даних. Ці ключі називаються відкритим ключем і закритим ключем. Публічний ключ, як випливає з назви, доступний усім, хто бажає надіслати повідомлення. З іншого боку, приватний ключ зберігається в безпечному місці власником відкритого ключа.

Публічний ключ шифрує інформацію, яку потрібно надіслати. При цьому використовується певний алгоритм. Оскільки, приватний ключ, який знаходиться у розпорядженні одержувача, розшифровує його. Один і той же алгоритм стоїть за обома цими процесами.

Залучення двох ключів робить асиметричне шифрування складною технікою. Таким чином, воно виявляється масовим вигідним з точки зору безпеки даних. Алгоритм Діффі-Хеллмана та RSA є найбільш широко використовуваними алгоритмами для асиметричного шифрування [4].

Асиметричний алгоритм більш захищений шляхом створення пари ключів, але і він має вразливі місця. Один зі шляхів усунення цих вразливих місць є використання динамічного ключа, тобто ключа, який змінюється на кожній ітерації.

Виходячи з цього, виникає задача знаходження захищеного алгоритму шифрування після аналізу існуючих за критеріями. Розробка або удосконалення алгоритмів в разі необхідності.

2.2 Математична модель

Для шифрування блоків даних при передачі між об’єктами доцільно використовувати асиметричний спосіб шифрування, для встановлення “Handshake” між вузлами каналу. “Handshake” досягається шляхом обміну між об’єктами публічними ключами. Криптосистеми відкритого ключа зручні, оскільки вони не вимагають, щоб відправник і одержувач мали загальний секрет для надійного зв'язку (серед інших корисних властивостей). Проте, вони часто покладаються на складні математичні обчислення, і, як наслідок, вони набагато менш ефективні, ніж криптосистеми з симетричними ключовими характеристиками. У багатьох

застосунках висока вартість шифрування довгих повідомлень у криптосистемі відкритих ключів може бути надмірною. Рішенням цієї проблеми, та для досягнення максимальної швидкодії, є використання гібридного шифрування, який поєднує в собі переваги симетричного та асиметричного шифрування.

Гібридна криптосистема може бути побудована за допомогою будь-яких двох окремих криптосистем:

- схема інкапсуляції ключів, яка є криптосистемою публічного ключа;
- схема інкапсуляції даних, яка є схемою симетричного ключа.

Гібридна криптосистема сама по собі є системою відкритих ключів. Загальнодоступні та приватні ключі яких подібні до схеми інкапсуляції ключів [5].

Для дуже довгих повідомлень основна частина роботи в шифруванні / дешифруванні здійснюється за допомогою більш ефективної схеми симетричних ключів, тоді як неефективна схема публічних ключів використовується лише для шифрування / розшифрування короткого значення ключа.

Всі практичні реалії криптографії публічного ключа сьогодні використовують гібридну систему. Приклади включають протокол TLS, який використовує механізм відкритих ключів для обміну ключами (наприклад, Diffie-Hellman) та механізм симетричного ключа для інкапсуляції даних (наприклад, AES). Інші приклади - формат файлу OpenPGP (RFC 4880) і формат файлу PKCS # 7 (RFC 2315) [6].
Гібридний алгоритм є досить сучасним і надійним, але залишається проблема статичного ключа. Якщо зловмисник має доступ до ключів, то нема перешкод в дешифровці. Тому необхідно додати в алгоритм систему зміни ключа – базовий динамічний ключ. Динамічний ключ має змінюватись на кожній ітерації.

Для розробки комплексного алгоритму, необхідно вирішити такі задачі:

- аналіз асиметричних алгоритмів і визначення найбільш підходящого для комплексного впровадження;
- аналіз симетричних алгоритмів і визначення одного або декількох найбільш підходящих для комплексного впровадження;
- розробка алгоритму динамічної зміни базового ключа;

- розробка комплексного алгоритму на основі гібридного поєднання симетричного та асиметричного алгоритмів з алгоритмом з динамічним ключем.

2.3 Аналіз асиметричних алгоритмів

Основним асиметричними алгоритмами виступають RSA, Deffie-Hellman, DSA, ECC.

RSA алгоритм

RSA є одним з методів шифрування відкритих ключів. Це був перший алгоритм, розроблений у криптографії з відкритим ключем, і одне з перших великих досягнень у шифруванні відкритих ключів. Він включає в себе три етапи:

ЕТАП 1. Генерація ключів
ЕТАП 2. Шифрування
ЕТАП 3. Дешифрування

А тепер розглянемо детально ці етапи:

ЕТАП 1: Для генерація ключів RSA використовуються два ключі для процесу. Шифрування здійснюється за допомогою відкритого ключа приймача, а дешифрування здійснюється за допомогою приватного ключа отримувача. Щоб створити ключ, використовуються наступні кроки:

КРОК 1. Вибираються два різних і великих простих числа, кажуть, P і Q.

КРОК 2. Обчислюється N такі, що, N = P * Q.

КРОК 3. Обчислюється z таким, що, z = (P-1) * (Q-1).

КРОК 4. Вибирається експонента публічного ключа: E така, що 1 <E <z, а E та z не мають жодних спільних дільників, крім 1.

КРОК 5. Визначається D, який задовольняє співвідношенню E * D = 1 (мод z). E ділиться на найменші з серій: z + 1, 2z + 1, 3z + 1, 4z + 1, . тд.

Тепер маємо відкритий ключ: (E, N) та приватний ключ: (D, N).

ЕТАП 2: Шифрування - це процес перетворення звичайного тексту в шифрований. Цей процес вимагає двох речей: ключа та алгоритму шифрування. Шифрування відбувається на стороні відправника, і використовується таке рівняння для шифрування повідомлення:

C = M ^ E mod (N),

де C - шифр тексту, а M - звичайний текст або повідомлення.

ЕТАП 3: Дешифрування - це процес перетворення шифрованого тексту в звичайний текст. Цей процес вимагає двох речей: алгоритм дешифрування та ключ. Дешифрування відбувається на стороні приймача, і для розшифрування повідомлення використовується таке рівняння; M = C ^ D mod (N). Процеси шифрування та дешифрування показані на рисунку 2.1 [7].

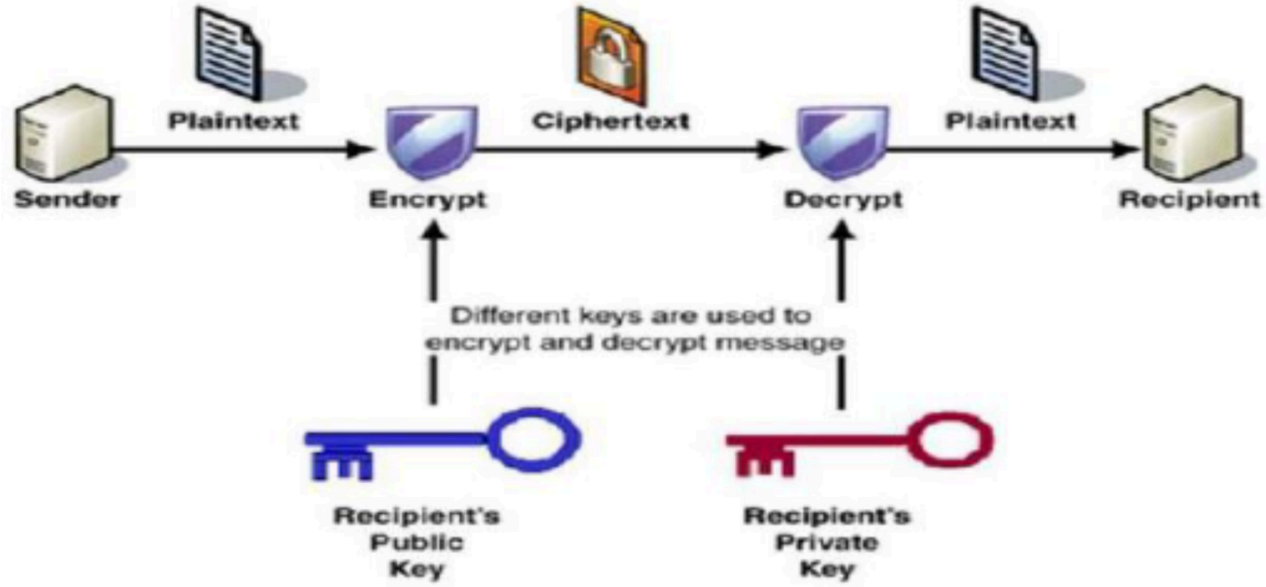


Рисунок 2.1 – Алгоритм RSA

Deffie-Hellman алгоритм

Алгоритм Діффі-Хеллмана дозволяє обидвам сторонам, які не знають попередньо один про одного, взаємно встановлювати загальний секретний ключ в небезпечному каналі. Обмін ключами Діффі-Хеллмана оснований на симетричній криптографії, оскільки загальний секретний ключ і ключ сеансу використовуються для шифрування та дешифрування. Алгоритм використовується багатьма протоколами, такими як SSL, Secure Shell та IPSec [8].

Кроки цього алгоритму такі:

КРОК 1. Виділяються два числа "p" (просте число) і "g".

КРОК 2. Вибираються два секретних числа "x" для відправника та "y" для приймача.

КРОК 3. Обчислюється загальнодоступне число R1 = g ^ x mod p, і R2 = g ^ y mod p.

КРОК 4. Здійснюється обмін ключами.

КРОК 5. Обчислюється перший сеансовий ключ як Ks, Ks = R2 ^ x mod p.

КРОК 6. Обчислюється другий сеансовий ключ як Kr, Kr – R1^y mod p

КРОК 7. тут Kr=Ks=K

Алгоритм представлений на рисунку 2.2 [9].

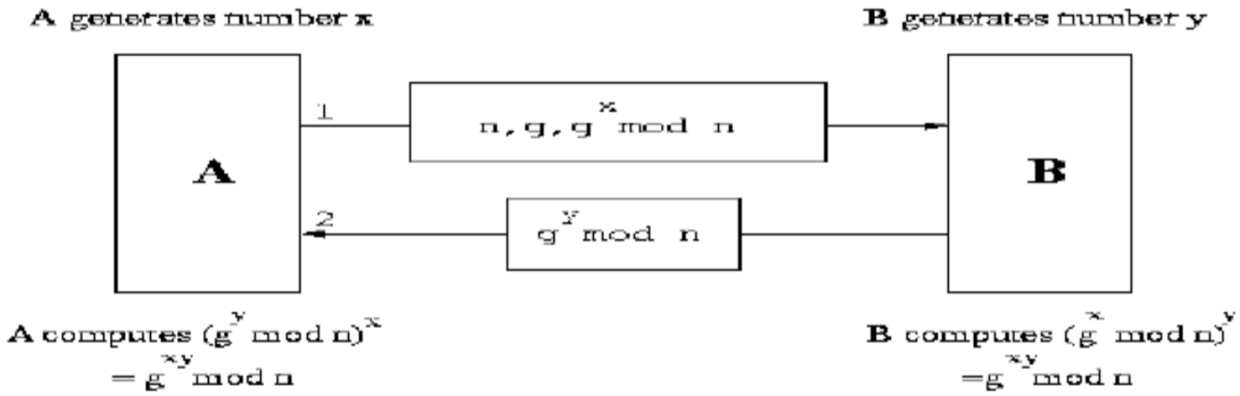


Рисунок 2.2 – Алгоритм Deffie-Hellman

Відправник А і Приймач В хоче поділитися своїми секретними ключами через небезпечний канал. Вони не діляться інформацією, вони просто діляться ключами. Основним недоліком цього алгоритму є те, що в цьому алгоритмі відбувається атака man-in-middle-attack. Це відбувається під час обміну загальнодоступними числами, тобто R1 і R2. Під час вторгнення змінюється значення R1 і R2 і передається нове обом сторонам. З цієї причини значення ключа сеансу стає нерівним.

Алгоритм DSA

Цифрові підписи є одним з кращих інструментів для забезпечення безпеки. Цифровий підпис - це електронна версія письмового підпису. Це криптографічний алгоритм відкритого ключа, який забезпечує автентифікацію, авторизацію. Цифровий сертифікат - це цифровий ідентифікатор, який показує ідентифікацію в мережі. Технологія шифрування як ядро цифрових сертифікатів може здійснювати шифрування та дешифрування, а також перевірку цифрового підпису та підпису інформації, переданої в мережі, для забезпечення цілісності конфіденційності та

безпеки інформації, що передається в Інтернеті. Цифровий підпис реалізується за допомогою алгоритмів публічного та приватного ключів та хеш-функцій. Він використовується на стороні приймача для підтвердження повідомлення та ідентифікації відправника. Весь процес цифрового підпису показаний на рисунку 2.3 [10].

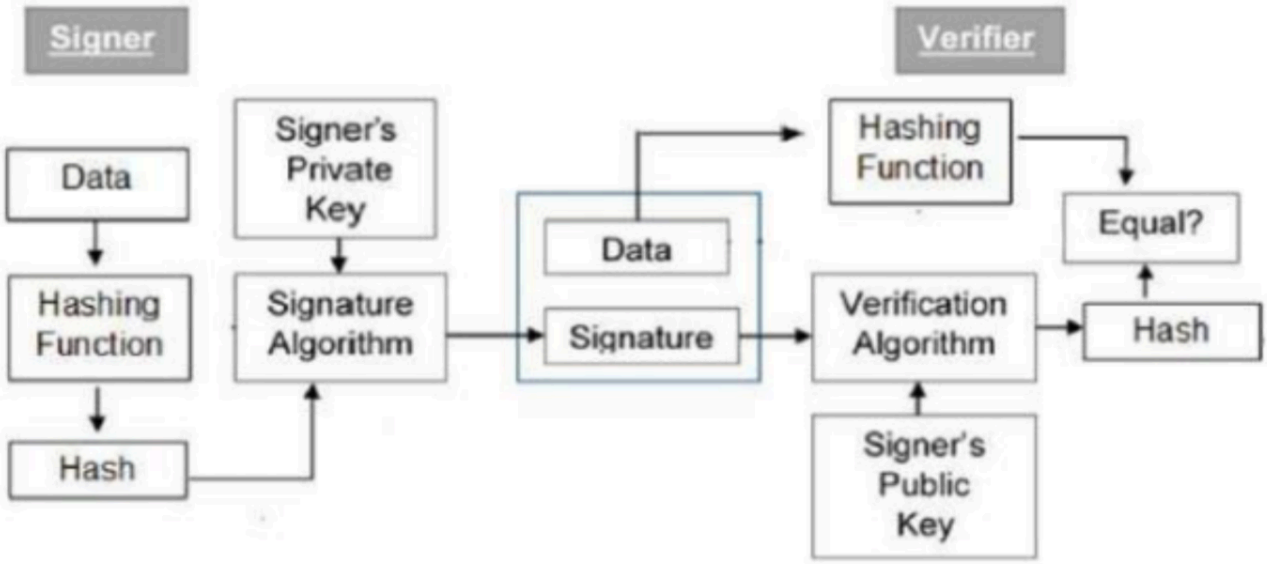


Рисунок 2.3 – Алгоритм DSA

Властивості Hash функції:

- функція хеша повинна знищувати всі гомоморфізмні структури в основній криптосистемі відкритого ключа (неможливо обчислити хеш-значення 2-х повідомлень, об'єднаних з урахуванням їх окремих хеш-показників);
- функція хеша повинна бути обчислена по всьому повідомленню;
- функція прив'язки повинна бути функцією в одну сторону, щоб повідомлення не розкривались підписом;
- хеш-функція має бути невираховною для іншого повідомлення з тим самим значенням хеша [10].

Цей алгоритм популярний для ".doc, .pdf, .txt" та для інших типів файлів. Функція хешу може використовуватися для динамічного розміру даних. Термін "динамічні засоби", результати хеш-функції залежать від розміру даних. Застосування цифрових підписів полягає в забезпеченні інформаційної безпеки (автентифікація, цілісність даних), електронної комерції, банківської справи,

розподілу програмного забезпечення та під юрисдикцією, а також для виявлення підробки або втручання в дані.

ECC алгоритм

Цей алгоритм в основному залежить від алгебраїчної структури еліптичних кривих. ECC включає в себе три етапи експлуатації, тобто ключове узгодження, шифрування та алгоритми цифрового підпису. Першим кроком є ключовий алгоритм розподілу, який використовується для спільного використання секретного ключа, другий крок - це алгоритм шифрування, який забезпечує конфіденційне спілкування, а останній - це алгоритм цифрового підпису, який використовується для автентифікації підписувача, тобто відправника, та перевірки цілісності повідомлення.

Еліптична крива є плоскою кривою яка повинна задовольняти наступним рівнянням: y2 = x3 + ax + b

ECC вважається кращим для створення більш швидких, і менших ефективних ключів . ECC пропонує еквівалентну кількість безпеки для набагато менших розмірів ключа, а отже, зменшує витрати на обробку та комунікацію. ECC вважається найбільш підходящим для сенсорних мереж, що забезпечує гарний компроміс між ключовими розмірами та безпекою [11].

Нижче наведені кроки для ECC:

КРОК 1. Користувач спочатку повинен кодувати будь-яке повідомлення M як точку на еліптичній кривій Pm.

КРОК 2. Вибирається відповідна криву і точку G, як у Diffie-Hellman.

КРОК 3. Кожен користувач вибирає приватний ключ nA <n і обчислює відкритий ключ PA = nAG.

КРОК 4. Для шифрування, шифрується Pm, Cm = {kG, Pm + kPb}, де k - випадкове число.

КРОК 5. Для розшифрування розшифрується Cm, обчислюючи: Pm + kPb-nB (kG) = Pm + k (nBG) -nB (kG) = Pm [12].

Таблиця 2.1 містить дані, які дозволяють провести порівняльний аналіз розглянутих алгоритмів.

Таблиця 2.1 – Порівнююча таблиця алгоритмів

Алгоритм	*Розмір ключа*	*Переваги*	*Недоліки*	*Можливі атаки*	*Контрзаходи для атак*
RSA	1024, 2048, 3072, 4096	Менший обчислювальний час	Малий показник шифрування та мале повідомлення. Той самий ключ для шифрування та підписання. Використання спільного модуля для різних користувачів	Атака Хастада, Франкліна-Рейтера, атака по частковій експоненті	Оптимальне асиметричне шифроване доповнення
Deffie-Hellman	1024, 3072	Вирішує складний дискретний логарифм. Створення і розповсюдження ключа, не інформації	Дорога експоненціальна операція. Відсутність автентифікації.	Атака Головний по центру	Використання алгоритму автентифікації з алгоритмом D-H

Алгоритм	*Розмір ключа*	*Переваги*	*Недоліки*	*Можливі атаки*	*Контрзаходи для атак*
DSA	від 512 до 1024 (включно)	Автентифікація Цілісність даних Невідхильність	Ентропія, секретність та унікальність випадкового значення є критичними.	Атака відновлення ключа	-
ECC	160, 224,56	Менший розмір ключа. Зниження рівня пам'яті. Зменшення часу передачі. В 15 разів швидше, ніж RSA. Менша споживана потужність	Збільшується розмір зашифрованого тексту. Залежить від дуже складних рівнянь, що підвищують складність алгоритму	Атаки стороннього каналу Backdoors. Квантові обчислювальні атаки	

Навантаження алгоритмів експериментами не має сенсу, оскільки алгоритми досить різні і мають різну сферу застосування. Алгоритм Deffie-Hellman побудований на симетричній базі, тому в гібридному використанні він не доцільний. Алгоритм DSA має застосовуватись більше для шифрування файлів і цифрового підпису.
Можна розглядати алгоритми ECC та RSA. По швидкодії алгоритм ECC має більше переваг, але розмір ключа занизькій. Тому для шифрування будемо використовувати RSA алгоритм.

2.4 Аналіз симетричних алгоритмів

В процесі аналізу будуть розглянуті алгоритми: DES, 3DES, AES, Blowfish

DES

DES не виконує вимогу RFC, використовуючи лише 56-бітні клавіші, що дає 256 можливих ключів. Цей відносно невеликий розмір ключа робить його вразливим до атак і багато разів було зламано, наприклад, за допомогою кластерного комп'ютера FPGA SciEngines FPGA, використовуючи 128 Spartan-3 5000 FPGA. Використовуючи пристрій грубої сили з швидкістю вгадування в тисячі мільярдів кнопок на секунду 1012 клавіш / сек, для пошуку ключа потрібно менше дня. Тому DES не вважається захищеним алгоритмом і його було видалено з TLS 1.2 [13].

3DES

Алгоритм 3DES використовує два або три ключа DES, які є 56-бітними і забезпечують безпеку від 56 біт до 168 біт залежно від параметра ключа. 3DES має три ключові параметри:

- варіант 1 - всі три ключі незалежні один від одного, що забезпечує найсильніший захист, що має ключ 3,66 = 168 біт;
- варіант 2 -два з трьох ключів є незалежними, надаючи цьому параметру захист 2 · 56 = 112 біт; цей розмір ключа є на кілька бітів менше, ніж рекомендований мінімум 128 біт;
- варіант 3 - тут всі ключі ідентичні, що еквівалентно DES, що надає лише 56-бітну клавішу безпеки; цей варіант, як згадувалося вище, більше не рекомендується Національним інститутом стандартів і технологій (NIST).

Використовуючи найменший можливий ключ безпеки (за винятком ключового варіанта 3), надається 2112 можливих варіантів, що займе 1.6 · 1014 років, щоб перевірити всі ключі з використанням швидкості 1012 кл / сек. Це можна вважати досить безпечним алгоритмом, але не вважається високо захищеним NIST. Навіть при довжині ключа 3DES - 168 біт, сила безпеки може бути зменшена до 112 біт лише тоді, коли використовується найбільш стандартна техніка атаки 3DES, яка

називається атакою Meet-In-The-Middle, яка вимагає 2112 кроків для шифрування [14].

Blowfish

Ключ змінної довжини для Blowfish може коливатися від 32 до 448 біт, за замовчуванням він має 128 біт, використовуючи 64-бітний блоковий шифр. Найменший захист, який може бути наданий, - це 232 можливих ключа до найвищої безпеки 2448 можливих ключів з періодом грубого застосування від кількох хвилин до 10115 років. Найвідоміший загальнодоступний криптоаналізм - це усічено-диференціальний-криптоаналіз або атаки слабких ключів, але на сьогодні не знайдено ефективного криптоаналізу. Blowfish ще не був зламаний і сьогодні вважається як безпечний алгоритм шифрування [15].

AES

AES має розмір ключів 128 біт, 192-бітних і 256-бітових, він здатний виробляти від 2128 до 2256 можливих ключів. Використовуючи найменшу безпеку AES, тобто 128-бітовий розмір ключа, знадобиться 1019 років, щоб знайти потрібний ключ, використовуючи швидкість 1012 ключів / сек. AES ще не був зламаний. Хоча найкращий публічний криптоаналізм - це атака бічного каналу для використання слабких місць у реалізації криптографічного алгоритму і, таким чином, не є суто пов'язаною з цією тезою в цьому контексті. На сьогоднішній день ніяка відома практична атака не порушила алгоритм AES, який вказує на те, що AES є безпечним [16].

Тест на швидкість шифрування.

Виконаємо порівняння працездатності між DES, 3DES, Blowfish та AES. Для вимірювання часу, візьмем ПК з процесорами Pentium-II 266 МГц і Pentium-4, 2,4 ГГц, на обох використовується операційна система Microsoft Windows. Платформа для використання - Java (JDK 1.4). Результати показані на рисунках 2.4 та 2.5.

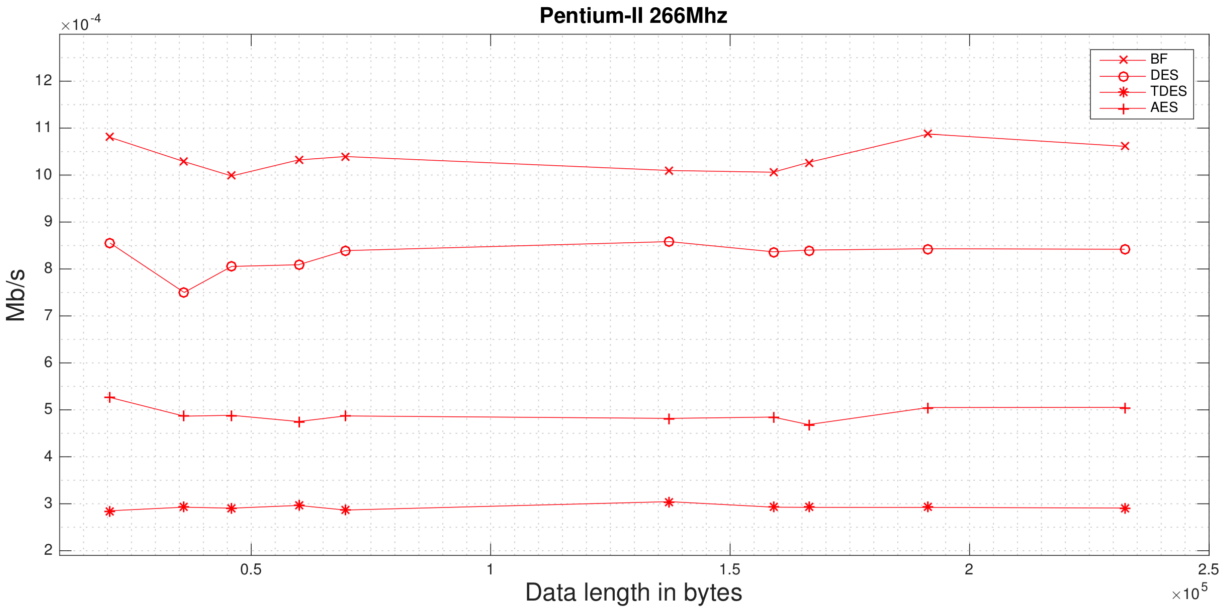


Рисунок 2.4 – Залежність швидкості виконання від розміру даних на Pentium-II 266-МГц

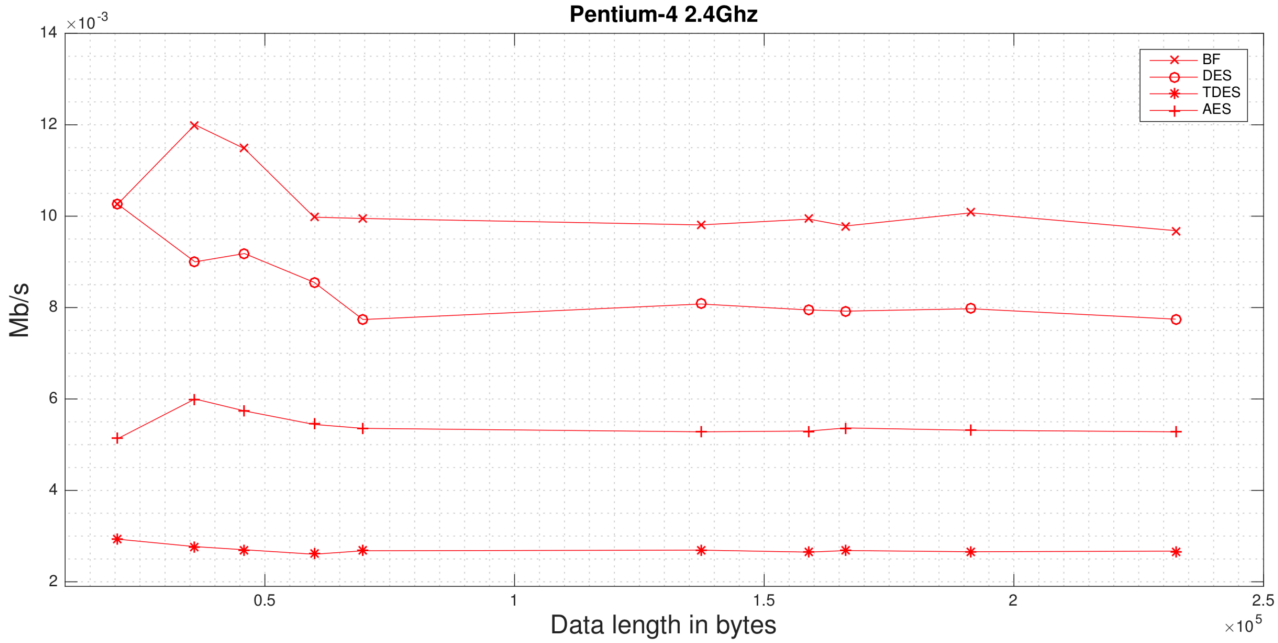


Рисунок 2.5 – Залежність швидкості виконання від розміру даних на на Pentium-4 2,4 ГГц

На рисунках бачимо, що середня пропускна здатність становить від 11 секунд до 383 секунд, що дуже повільно.

Виходячи з рисунків, можна зробити висновок, що продуктивність 3DES дійсно погана, у порівнянні з AES, DES і Blowfish. DES майже у два рази швидше, ніж AES, але все ще повільніше, ніж Blowfish.

В дослідженні Гурприта Сінгха, два інших блокових шифрів DES і AES мають аналогічну швидкість роботи. Blowfish є найшвидшим для цих розмірів даних [11]. Причина низької швидкодії 3DES може бути пов'язана з тим, що алгоритм повинен виконувати три ітерації звичайного алгоритму DES, щоб відповідати стандартам безпеки, які DES не виконує.

Тимо Бінгманн опублікував блог 14 липня 2008 р. під назвою "Швидкість тестування та порівняння відкритих вихідних кодів бібліотек та компілювальних прапорів". Там він порівняв багато відомих бібліотек з криптографією, на процесорі Pentium 4 на частоті 3,2 ГГц. У цьому дослідженні, OpenSSL, попередник ARM з TLS, є однією з порівнюваних бібліотек. Він порівнює пропускну здатність AES (Rijndael), Blowfish, CAST5 і 3DES. Слід зауважити, що CAST5 - це шифр, який не входить до статті, і не буде розглядатися далі [15].

На рисунку 2.6 показано, як довжина зашифрованих даних впливає на швидкість та продуктивність алгоритму шифрування. Це пов'язано з ключовим періодом попередньої обробки / ініціалізації, що стає менш ефективним у випадку більшого розміру буфера. Розміри ключа використовувались стандартні, хоч і не були зазначені: (128 біт), DES (56-біт), 3DES (168-біт) і Blowfish (128-біт).

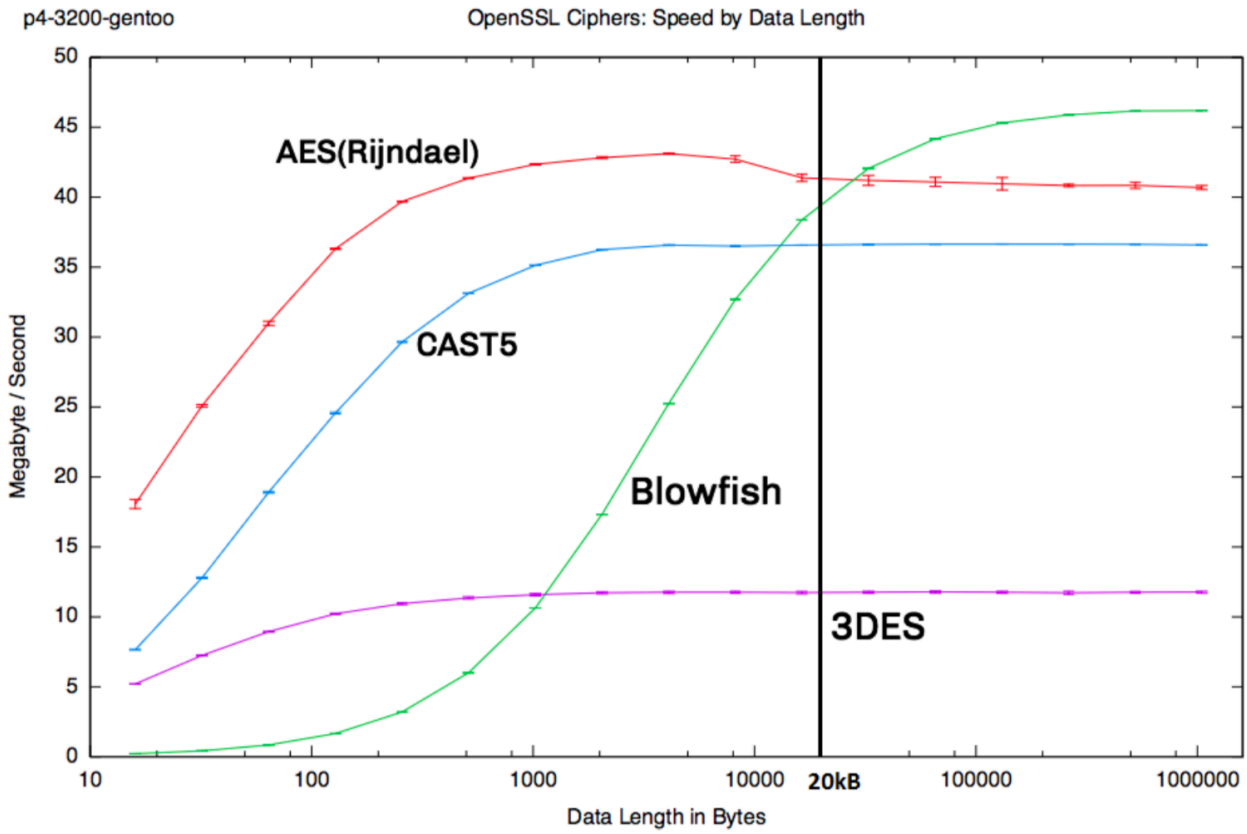


Рисунок 2.6 – Залежність продуктивності від довжини зашифрованих даних на OpenSSL [15]

Порівнюючи швидкодію алгоритмів у бібліотеках шифрування, чудово видно, як вона масштабується з різними можливостями апаратного або центрального процесора. Використовуючи бібліотеку OpenSSL на комп'ютері Intel Pentium 2, 300 МГц, можна провести наступні тести, результати яких, зображені на рисунку 2.7.

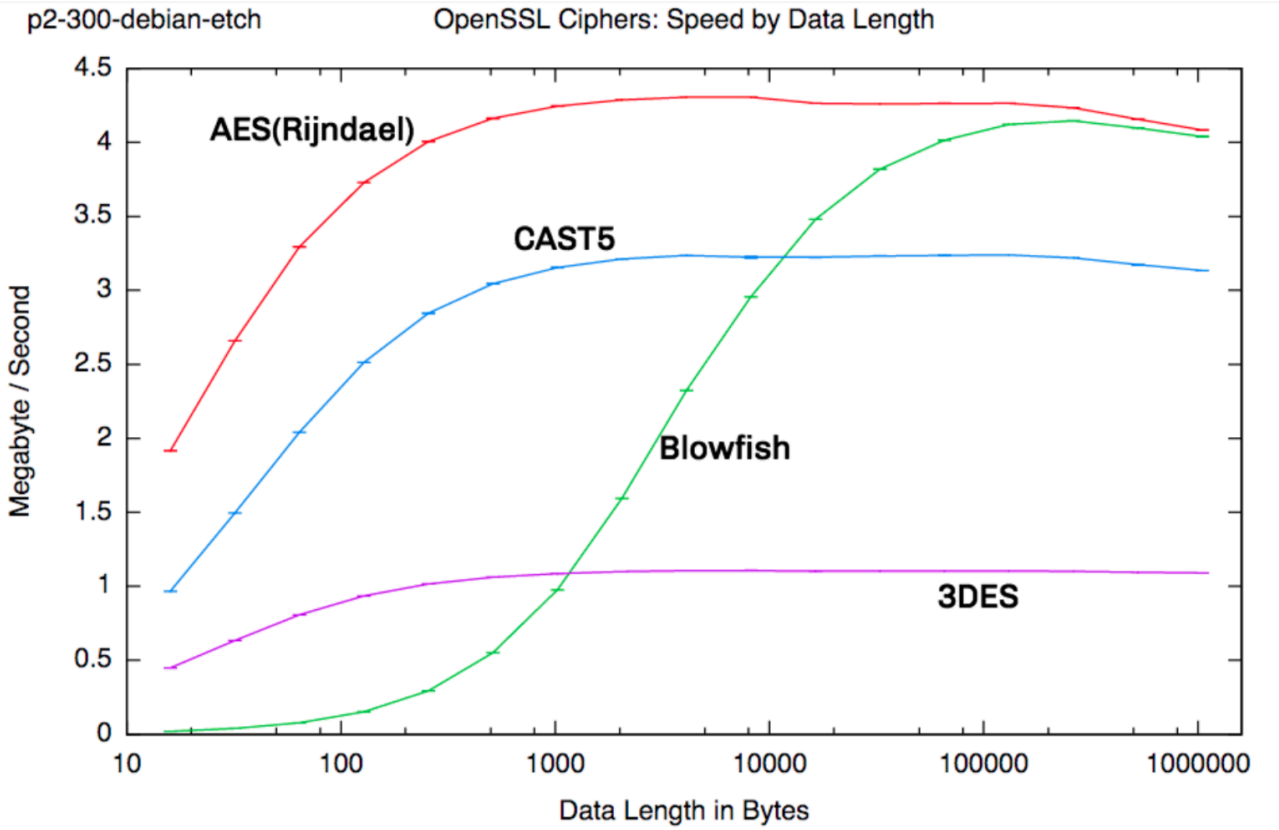


Рисунок 2.7 – Залежність продуктивності від довжини зашифрованих даних на OpenSSL (Intel Pentium 2 при 300Mhz) [15]

Можна зробити висновок, що в середньому, для довгих даних Blowfish швидше, ніж AES, DES і 3DES. Але для довжин менше ніж 20kB пакетів, AES є найшвидшим алгоритмом шифрування в цій конкретній бібліотеці.

Налаштування ключа

У таблиці 2.2, наведені результати алгоритмів з бібліотеки Crypto++ [17]. Результати були отримані за допомогою Microsoft Visual C ++, на процесорі Intel Core 2 1,83 ГГц.

Таблиця 2.2 – Швидкість ініціалізації ключа

Алгоритм	*Ініціалізація ключа*	*Цикли ініціалізації*
AES-CTR-128-bit	0.698	1277
DES-CTR	8.309	15320
3DES-CTR	27.317	49989

Алгоритм	*Ініціалізація ключа*	*Цикли ініціалізації*
Blowfish-CTR	62.683	114710

Тут AES значно швидше, ніж DES, 3DES і Blowfish в процесі налаштування ключа в блочному CTR. Це важливо, коли ініціалізація ключа є частою дією. Основна причина, по якій Blowfish робить налаштування початкового ключа доволі повільною операцією, полягає у тому, що це ускладнює атаку грубої сили (key-endfall).

Простір складності

У дослідженні, написаному Асіфом Муштаком та трьома іншими, було проведено порівняння з алгоритмом DES, 3DES, AES та Blowfish на критерій використання простору. Там він досліджує, як збільшиться розмір відкритого тексту після його шифрування, за допомогою різних алгоритмів. Звичайний текст, розміром 240 КБ, зашифровували кожним алгоритмом шифрування. Порівнювали розмір вироблених шифрованих текстів, результати наведені в таблиці 2.3 [18].

Таблиця 2.3 – Простір складності

Алгоритм	Текст	Шифрований	Дешифрований
DES	240KB	328KB	240KB
3DES	240KB	614KB	240KB
AES	240KB	847KB	240KB
Blowfish	240KB	955KB	240KB

З результатів цього експерименту видно, що алгоритм DES використовує найменшу кількість простору в порівнянні з іншими алгоритмами. Blowfish вимагає трохи більше місця для шифрування тексту, ніж алгоритм AES, що може стати вирішальним фактором в обсязі даних, що відправляються між клієнтом і сервером. Крім того, якщо клієнт не має доступу до серверу і зберігає вхідні зашифровані дані доки доступ не з’явиться, в умовах обмеженої кількості пам'яті, додаткове використання простору алгоритмом є не доречним.

Розмір ключа

Для всіх алгоритмів, розмір ключа впливає на швидкість виконання. За допомогою великих ключів зменшується швидкість виконання. У дослідженні Вей Дая досліджується швидкість AES з різними розмірами ключа, результати наведені в таблиці 3 [17].

Таблиця 2.4 – Швидкість алгоритму AES

Шифр	*Ключ*	*Циклів в біт*	*Мб/сек*
AES/CTR	128-bits	12.6	139
AES/CTR	192-bits	15.4	113
AES/CTR	256-bits	18.2	96
AES/CBC	128-bits	16.0	109
AES/CBC	192-bits	18.9	92
AES/CBC	256-bits	21.7	80

Дивлячись на результати видно, що збільшення розміру ключа зменшить швидкість роботи алгоритму. Навіть якщо результати відрізняються не декілька циклів, це сильно вплине на швидкість шифрування пакета даних з використанням великої кількості байтів.

В таблиці 2.5 зведені результати досліджень.

Таблиця 2.5 – Результати досліджень

Параметр	*DES*	*3DES*	*AES*	*Blowfish*
Розмір ключа	56	56, 112 або 168	128, 196 або 256	32-448
Тип	Блочний шифр	Блочний шифр	Блочний шифр	Блочний шифр
Розміри блоків	64	64	128	64

Параметр	*DES*	*3DES*	*AES*	*Blowfish*
Число раундів	16	48	10,12 або 14	16
Рік випуску	1975	1998	1998	1993
Brute Force Атака	Менш як день	Менш як день до $\approx 10^{34}$ років	10^{19} років до $\approx 10^{58}$ років	Пару хвилин до $\approx 10^{115}$ років
Більш ефективні атаки	Диференційований криптоаналіз	Meet in the middle атака	Паралельний канал	Слабий ключ
Зламаний	Так	Ні	Ні	Ні
Статус безпеки	Небезпечний	Частково	Безпечний	Безпечний
Швидкість шифрування	Швидкий	Повільний	Найшвидший	Повільний
Ініціалізація ключа	Середня	Повільна	Швидка	Найповільніша
Простір складності	Чудовий	Добрий	Добрий	Поганий

Виходячи з даних таблиці 2.5, можна зробити висновок, що доцільніше використовувати алгоритм AES, оскільки він має найкращу швидкість при досить великому розмірі ключа. Також простір складності є добрим при найшвидшій ініціалізації ключа. Варто відзначити, що AES добре документований, що дозволяє подальшу оптимізацію алгоритму. AES був обраний як стандарт шифрування. Сьогодні існує багато систем, що мають апаратне прискорення для AES, це є великою перевагою, бо може суттєво зменшити час виконання алгоритму [11].

2.4 Розробка алгоритму динамічної зміни базового ключа

Для рішення задачі впровадження динамічного ключа було вирішено розробити алгоритм з механізмом зворотного зв'язку. Кожен блок відкритого тексту або частини (крім першого) побітово складається по модулю 2 (операція XOR) з попереднім результатом дешифрування. Таким чином дані будуть зашифровані по різному на кожній ітерації, а для дешифрування потрібно мати відкритий текст з попередньої ітерації.

Процес шифрування проілюстровано на рисунку 1 та може бути описано наступним чином:

$$C_0 = V \tag{2.1}$$

$$C_i = E_k(P_i \oplus P_{i-1}) \tag{2.2}$$

де i - номер блоку, V - вектор ініціалізації, C_i і P_i - блоки зашифрованого і відкритого текстів відповідно, а E_k - функція блочного шифрування.

Процес розшифрування продемонстрований на рисунку 2, в його основу покладено така операція:

$$P_i = C_{i-1} \oplus D_k(C_i) \tag{2.3}$$

де D_k, функція блочного розшифрування.

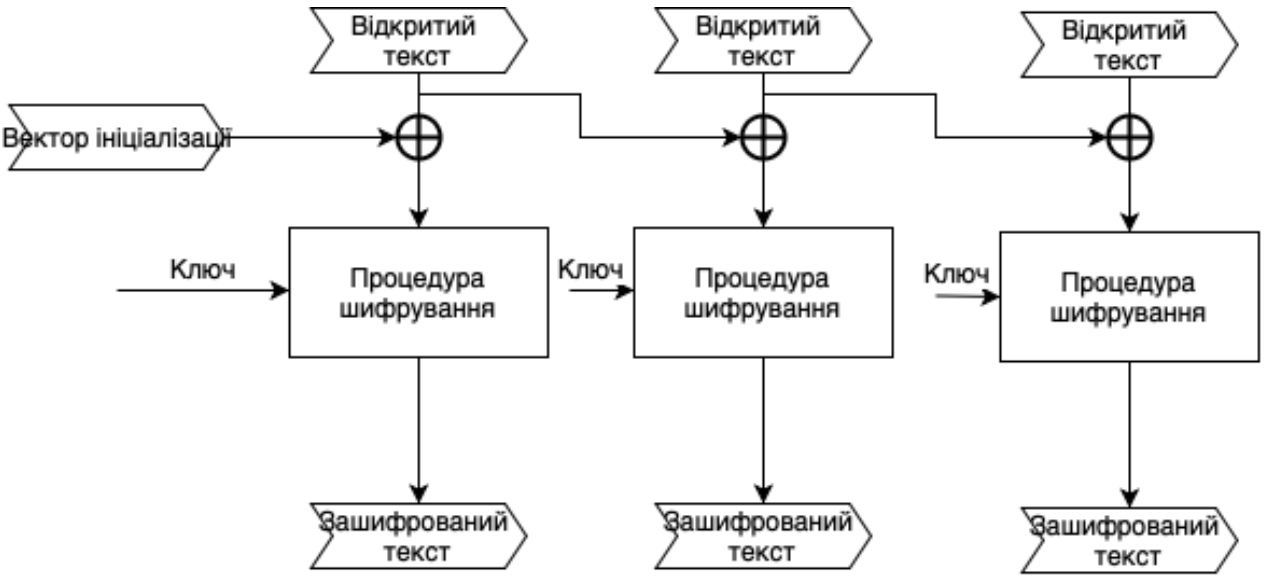


Рисунок 2.8 – Алгоритм шифрування

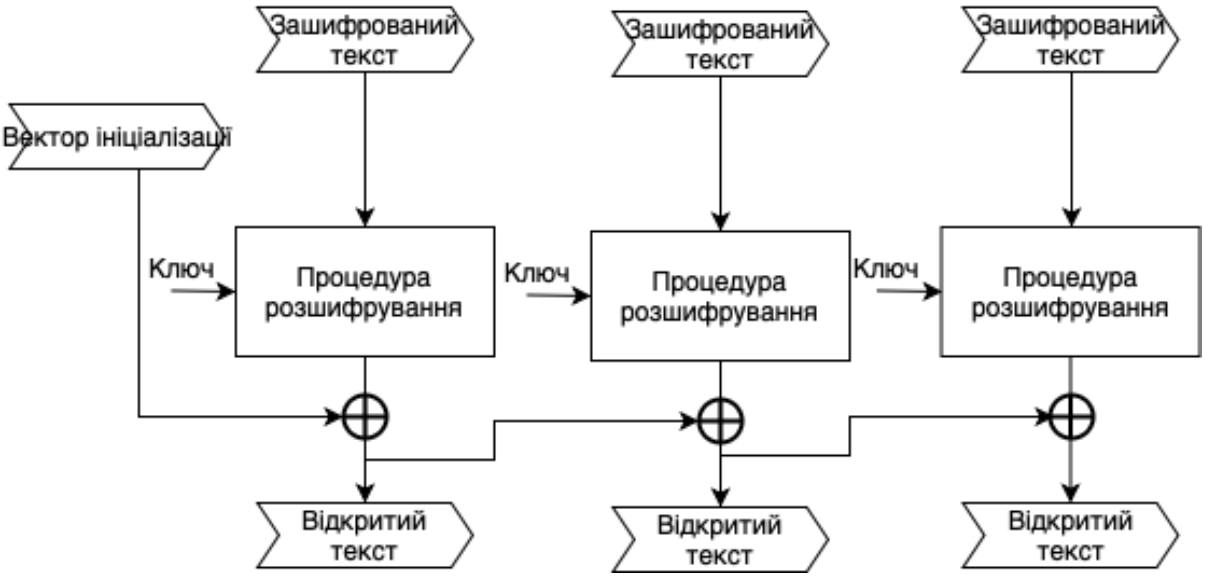


Рисунок 2.9 – Алгоритм дешифрування

Таким чином створюється деякий динамічний ключ, який передається в середині історії блоків. Швидкість операції XOR є настільки малою, що нею можна нехтувати. Для підвищення захисту операцію XOR можна замінити більш складною функцією, але це тягне за собою і збільшення часу операції.

2.5 Розробка комплексного алгоритму з динамічним ключем

Комплексний алгоритм базується на гібридному поєднанні симетричного, асиметричного алгоритмі та алгоритму з механізмом зворотного зв’язку. В якості симетричного алгоритму використовується алгоритм AES, в якості асиметричного – RSA.

Схема алгоритму шифрування:

КРОК 1. Відкритий текст побітово складається по модулю 2 з попереднім відкритим текстом (в першій ітерації – з вектором ініціалізації).

КРОК 2. Генерується новий ключ для шифрування AES і ним шифрується складений текст**.**

КРОК 3. Ключ вирівнюється до 256 біт і кладеться на початок зашифрованих даних.

КРОК 4. Ключ разом з шифрованим текстом шифрується публічним ключем RSA.

Схема відображена на рисунку 2.10.

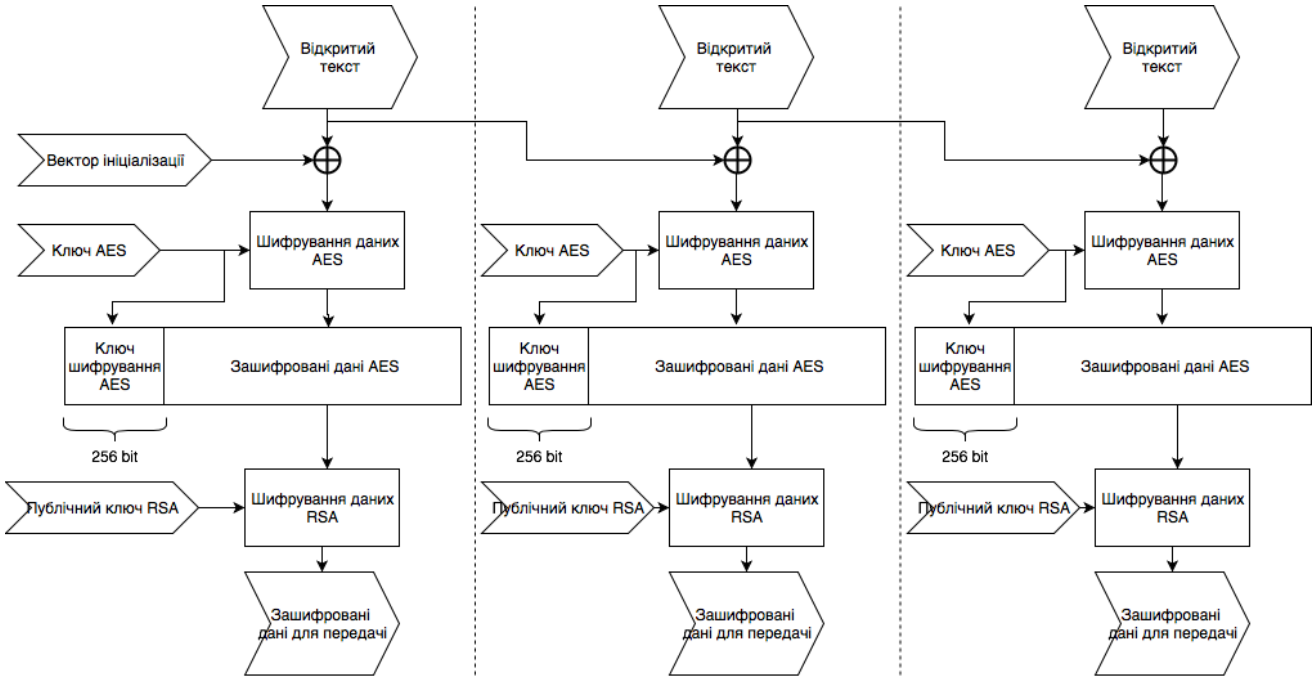


Рисунок 2.10 – Комплексний алгоритм шифрування

Схема алгоритму дешифрування:

КРОК 1. Зашифровані дані розшифровуються закритим ключем RSA.

КРОК 2. З початку розшифрованих даних вилучається 256 біт – ключ для AES.

КРОК 3. Данні без перших 256 біт розшифровуються AES алгоритмом.

КРОК 4. Розшифровані дані складаються по модулю 2 з попереднім відкритим текстом (для першої ітерації з вектором ініціалізації)

Схема відображена на рисунку 2.11

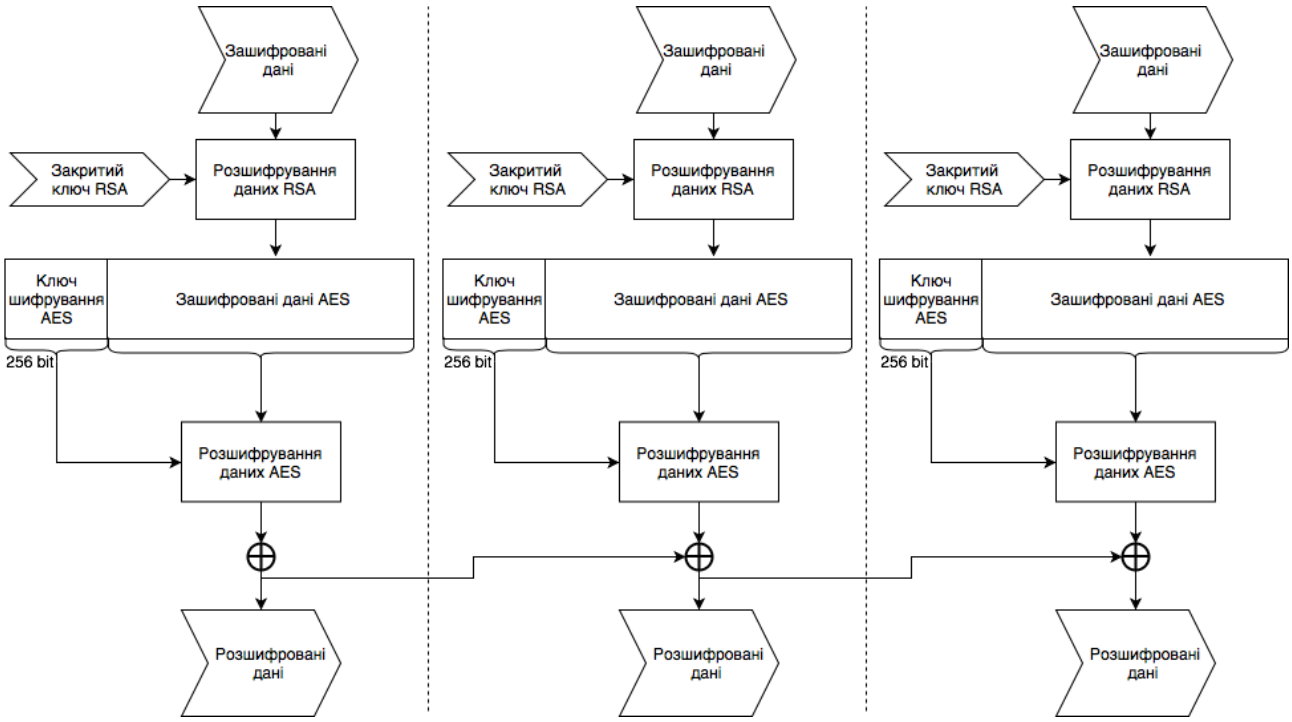


Рисунок 2.11 – Комплексний алгоритм дешифрування

Висновки до розділу

В цьому розділі була вирішена задача розробки комплексного асиметричного алгоритму з динамічним ключем. Були проаналізовані існуючі симетричні та асиметричні алгоритми та обґрунтовано вибір тих, що якнайкраще підходять до використанні в комплексі. Вирішення поставленої задачі було досягнене шляхом розробки гібридного алгоритму, що базується на обраних симетричних та асиметричних підходів, та використанні динамічного ключа. Задача введення динамічного ключа була вирішена шляхом створення алгоритму зі змінним ключем в залежності від даних. Була наведена схема та описана робота повного комплексного алгоритму. Використання цього алгоритму дозволить підвищити складність зламу та захищеність каналу, при передачі даних. Матеріалі аналізу були висвітлені в статті [19], розробка комплексного алгоритму в тезах [20].

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Програмний клієнт системи буде виходити в реліз на декількох платформа, для різних типів пристроїв.

Для персональних комп’ютерів, планується розробка під операційні системи macOS та Windows. Розробка під Linux, Google Chrome OS та ін. не має рентабельності. Однією з головних причин того, чого ці платформи не захопили велику частку ринку, полягає в тому, що більшість загального програмного забезпечення все ще створюється лише для платформ Windows та macOS. Хоча можна стверджувати, що платформи Linux та ін. мають деякі (іноді навіть кращі) альтернативи, основним фактом залишається те, що користувачі ПК використовують найпоширеніші, доступні та найпростіші у використанні платформи. Наразі ці ідеали представляють лише платформи macOS та Windows.

Для смартфонів та планшетів, планується розробка під операційні системи iOS та Android. Зараз, ці дві платформи, займають майже повністю весь ринок мобільних застосунків, і розглядати інші платформи нема потреби.

За останні роки мобільні операційні системи прогресують, перекриваючі потреби у використанні персональних комп’ютерів. Персональний комп’ютер стає пристроєм для роботи. Тому програмний клієнт для мобільних пристроїв більш важливий у реалізації.

На прикладі одного компонента у складі складної системи розглянемо засоби та технології, завдяки яким, були запрограмовані і вирішені задачі системи. Компонент представлений для мобільних пристрої, так як має найбільший пріоритет. Компонент був повністю розроблений і доведений до стану дистрибуції до кінцевого користувача.

3.1.1 Платформа розробки

Компонент – клієнт для мобільних застосунків написаний на платформі iOS. Ця платформа має багато якостей, декілька з них можна відмітити.

Якість UI та UX системи. Через те що у iPhone та iPad хоч екрани всі різних розмірів, вони всі пропорційні, що дозволяє робити якісний інтерфейс; також Xcode дозволяє розроблювати UI під всі види екранів;

Тривала підтримка старих пристроїв. Apple намагається до останнього підтримати старі пристрої. Для статистики, близько 90% користувачів iOS використовують останню платформу. Такий великий відсоток досягається тим, що, наприклад, остання iOS 12 підтримується на iPhone 5S, якому вже більше ніж 5 років. iPhone наразі рахується надійнішим смартфоном на ринку. Strategy Analytics дослідили що iPhone надійніше Samsung в три рази а Nokia у п’ять. Apple роками відточує виробництво однієї еко-системи, одного смартфону та однієї операційної системи.

Безпека. Apple дуже піклується про безпеку персональних даних, компанія постійно покращує засоби захисту особливо з Touch ID та FaceID в iPhone X. Apple має дуже високий рівень перегляду застосунків які виходять у реліз. Компанія також шифрує дані в iMessage та інших додатках. Apple надає пріоритет конфіденційності користувачів, тому вони можуть відчувати себе в безпеці, знаючи що дані зберігаються в зашифрованому вигляді.

3.1.2 Серидовище розробки

Для середовища розробки (IDE) був обраний XCode виробництва Apple.
Xcode - це середовище розробки для операційних систем Apple, таких як OS X, iOS, WatchOS і tvOS.

Інтегроване середовище розробки (IDE) містить набір інструментів для розробки програмного забезпечення, що підтримують вихідні коди для мов програмування. Xcode підтримує такі мови програмування як C, C ++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit (Rez) та Swift.

Розробники застосунків можуть створювати та редагувати програми, а також керувати програмою від ідеї, до випуску в App Store.

Xcode був вперше випущений в 2003 році. Компанія Apple випускає регулярні оновлення Xcode, оскільки виходять нові версії для власного програмного забезпечення [21].

Що стосується аналогів для написання під iOS існує тільки AppCoda. Але вона не дозволяє використовувати всі можливості при написання. Також викладати (дестреб'ютити) в AppStore - AppCoda не дозволяє, тому вибір був очевидний.

3.1.2 Мова програмування

Основні мови написання під iOS дві: Objective-C та Swift, для розробки була обрана перша.

Objective-C був створений під впливом двох інших мов програмування: C і Smalltalk. Ось чому він має такий складний, багатослівний синтаксис. Він отримав синтаксис об'єкта від Smalltalk, тоді як синтаксис для не об'єктно-орієнтованих операцій від C. Objective-C використовує динамічне типізацію та тип викликів функцій через передачу повідомлень. Це також вимагає розділення класів на два блоки коду: інтерфейс і реалізація.

Objective-C має деякі переваги:

Динамічна типізація. У деяких випадках, це дійсно може стати ключовою перевагою. Наприклад, спрощує створення нескладних програм;

Документація і спільноти. Більше 20 років успішного застосування мови посприяли появі великої кількості якісних ресурсів і книг. Сьогодні будь-хто, хто бажає вивчити Objective-C, без зусиль знайде відповідь на питання, що цікавить на просторах інтернету; у порівнянні з Swift 3.0, та й багатьма іншими мовами, Objective-C надає розробнику куди більше гнучкості;

Підтримка C/C++. Хороша сумісність з C і C ++. Оскільки Objective-C є суперсетією C, і, таким чином, з C або C ++, код працює досить гладко.

Стабільність. Якщо ви розробляєте програму в Objective-C, вам, імовірно, не доведеться витрачати гроші на перенесення додатка на нову мовну версію через кілька місяців [22].

Розвиток у Swift відбувається швидше, але це не єдиний фактор, який слід враховувати при прийнятті важливих бізнес-рішень.

Swift має ряд недоліків:

Використання сторонніх бібліотек. Застосунки та фреймворки все ще спираються на Objective-C;

Стабільність. Мова Swift все ще не є стабільною, тому велика частість оновлень можуть зашкодити проекту;

Потреба у Objective-C. через молодість мови і не перекладених на Swift кодів, OS X і iOS потрібно хоча б мінімальне знання Objective-C все однак;

Проблема компіляції. компілятор на Swift видає зайві помилки які просто збивають з пантелику.

3.1.3 База даних

Для проекту у якості бази даних була обрана Core Data на основі SQLite. Core Data - один з найпопулярніших фреймворків, наданих Apple для застосунків iOS та macOS. Core Data використовуються для керування модельним прошарком в програмі. Core Data розглядається як основа для збереження, відстеження, модифікації та фільтрування даних у застосунках, однак Core Data не є базою даних. За замовченням Core Data використовує SQLite, оскільки це “persistent store”, але сам фреймворк не є базою даних. Core Data має набагато більше функціональності ніж звичайна база даних, таких як управління графіками об'єктів, відстеження змін у даних та багато інших речей [23].

Core Data має ряд переваг у використанні:

- високий рівень керування пам'яттю. Core Data відвантажує в пам’ять лише ті об'єкти, які зараз використовуються; дані про об’єкти не відвантажуються в пам’ять до тих пір поки не будуть затребувані;

- коли існують зміни у наборі, зберігаються лише змінені об'єкти, а не весь набір даних;
- можна читати / писати об'єкти моделі безпосередньо, а не перетворювати їх на / з щось як наприклад асоціативний масив;
- вбудовано сортування об'єктів, при виборці їх із сховища даних;
- багата система предикатів для пошуку даних;
- відносини між сутностями обробляються безпосередньо як властивості пов'язаних об'єктів;
- необов'язкова автоматична перевірка значень властивостей;
- моделі даних не використовують масиви, але зв'язки "до багатьох" моделюються як набори.

3.2 Архітектура програмного забезпечення

Створена програма є iOS-застосунком, що не взаємодіє з фреймворками, в цілях безпеки.

Основний архітектурний шаблон – Viper. Viper - це шаблон дизайну, який реалізує парадигму "розділення концерну". По суті, як MVP або MVC випливає з модульного підходу. Одна функція, один модуль. Для кожного модуля VIPER має п'ять (іноді чотирьох) різних класів з різними ролями. Жоден клас не виходить за межі своєї єдиної мети. Ці класи слідують.

Вид (View): це код, який відображає інтерфейс для користувача та отримує їхню відповідь. Отримавши відповідь, сповіщає ведучого.

Представник (Presenter): Ядро модуля. Він отримує відповідь користувача з виду і відповідно опрацьовує, та передає усіма іншим компонентам. Закликає маршрутизатор для відображення інших модулів, Інтерактор для отримання даних (мережеві запити або локальні), контролює оновлення інтерфейсу користувача.

Інтерактор (Interactor): містить бізнес-логіку додатка. Перш за все робить виклики API, щоб отримати дані з джерела. Відповідальний за здійснення запитів даних, але не обов'язково від себе.

Маршрутизатор (Router): виконує дії від представника про те, який екран потрібно представити.

Сутність (Entity): містить класичні моделі, які використовує Інтерактор.

Модулі чату та трансферу даних побудовані на реактивному програмуванні, для швидкої передачі даних. Для цього використовується ReactiveCocoa (RAC) — це framework для компонування і перетворення послідовностей значень.

3.2.1 Діаграма послідовностей

На рисунку 3.1 зображена діаграма послідовності створення чат каналу та відправка повідомлення. Сутностями виступають контроллер відображення, прослофка для комунікації з сервером контролерами, та модуль який тримає Socket канал.

На перших етапах виконується встановлення Handshake між користувачами для формування захищеного каналу, на наступних етапах виконується відправка захищених повідомлень, шляхом вставка в потік Socket.

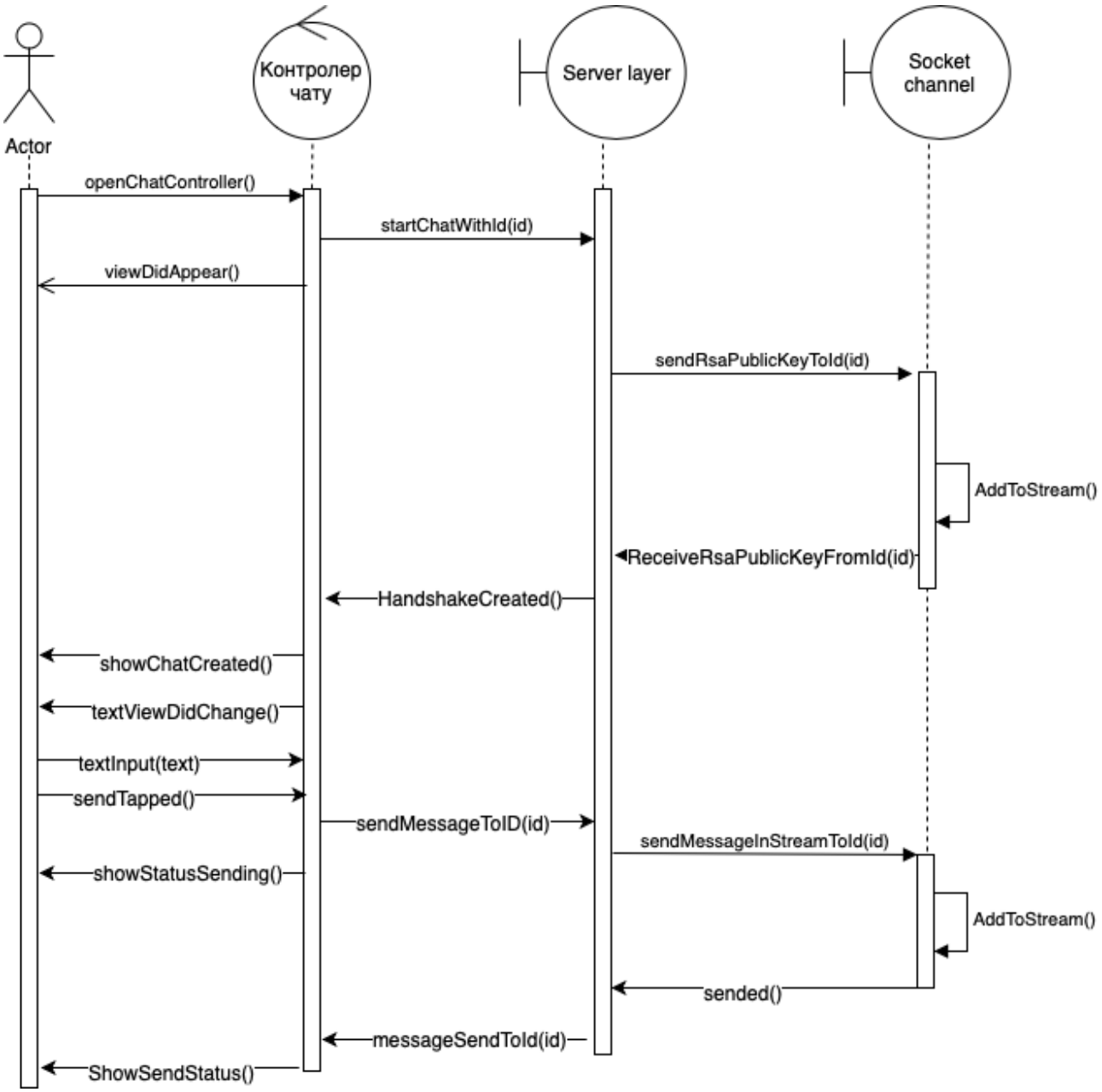


Рисунок 3.1 – Діаграма послідовності функціонування чату

3.2.2 Діаграма компонентів

На рисунку 3.2, зображені основні компоненти сервісу та взаємодія між ними. Згруповані модулі представлення та прошарки для комунікації між екранами та низькорівневими сервісами, такими як потокове відправлення повідомлень.

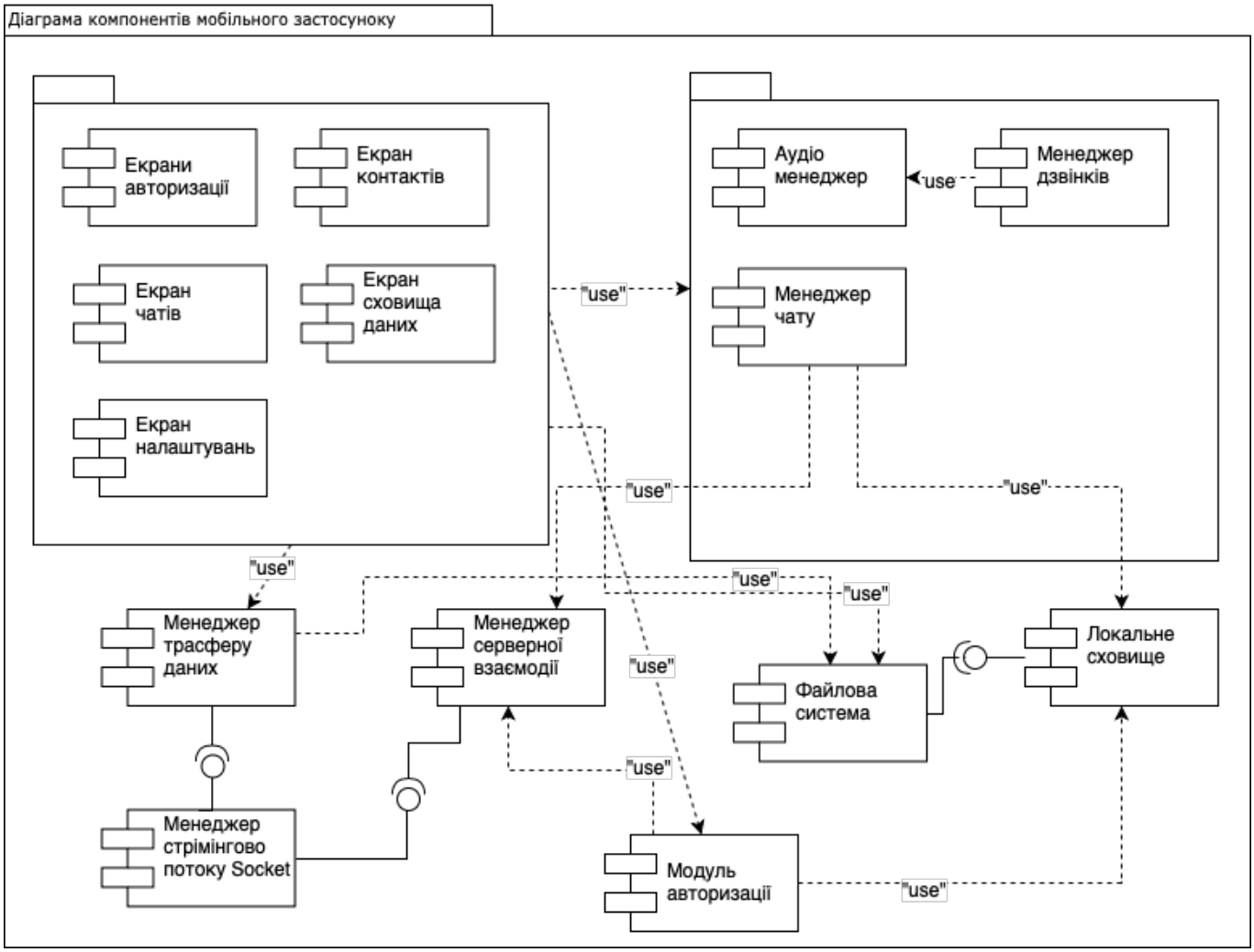


Рисунок 3.2 – Діаграма компонентів мобільного застосунку

3.3 Інструкція користувача

Застосунок на платформі iOS можна завантажити з AppStore, у будь-якій країні.

По-перше треба авторизуватись в системі. Застосунок нас зустрічає екраном введення мобільного телефону або пошти. Якщо аккаунт системи вже існує, то після введення номеру або пошти, треба ввести пароль і авторизуватись в системі. У разі відсутності аккаунту, треба пройти реєстрацію.

Реєстрація в системі займає три кроки. Першим кроком є підтвердження телефоного номеру або пошти, шляхом введення OTP коду з SMS або листа. Другим кроком є створення паролю і підтвердження його. Останнім кроком є введення інформації про себе.

Після проходження авторизації відкривається екран переписок з користувачами, а також головне меню внизу екрану. Також, показуються оповіщення про надання дозволу для відправки сповіщень та аналізу контактів в телефоні.

На першому пункті меню знаходяться контакти (Рисунок 3.3). При переході на цей екран, телефонні контакти аналізуються, для знаходження контактів в системі.

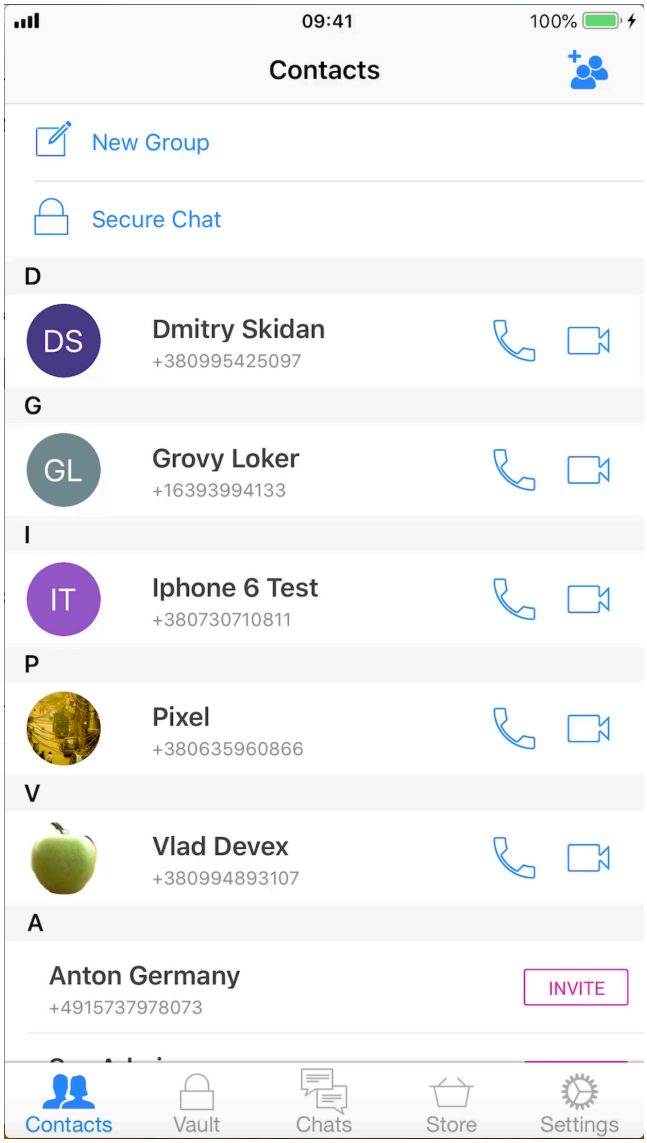


Рисунок 3.3 – Екран контактів

З контактами можна створювати дзвінки, як аудіо так і відео, а також створити захищений чат для обміну повідомленнями або файлами. Для створення чату з одним користувачем, достатньо натиснути на всю комірку в таблиці, для створення дзвінків – відповідні кнопки на комірці. Існує можливість створення групового захищеного чату, відповідна кнопка знаходиться в верхньому лівому куту екрану.

Наступним пунктом меню є доступ до сховища. Якщо сховище відкривається вперше, то з’явиться форма створення паролю для захисту (Рисунок 3.4).

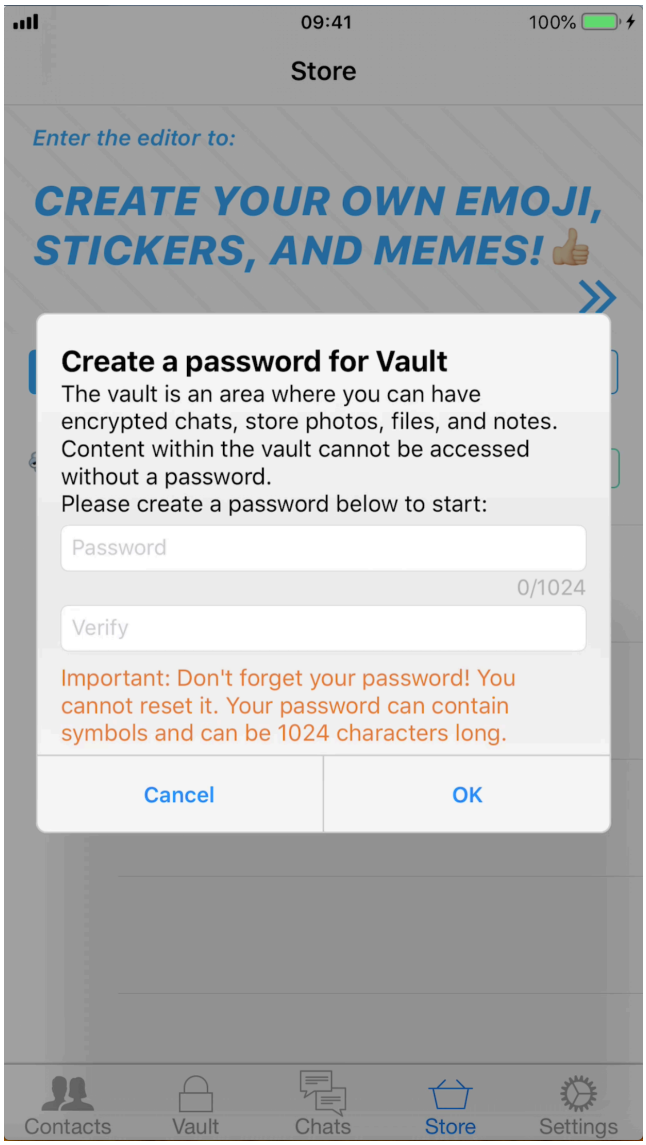


Рисунок 3.4 – Форма створення паролю для захисту сховища

Після створення паролю, локальне сховище відкрито до взаємодії. Сховище складається з трьох пунктів: чати, галерея та файли (Рисунок 3.5).

У разі надходження повідомлення, до значка меню, який відповідає за сховище, додається маркер з кількістю повідомлень (Рисунок 3.5). Повідомлення, разом з відправником виділяються жирним шрифтом, для позначення що повідомлення не прочитано. Поруч з повідомленням кольором позначений статус каналу передачі даних (Рисунок 3.5).

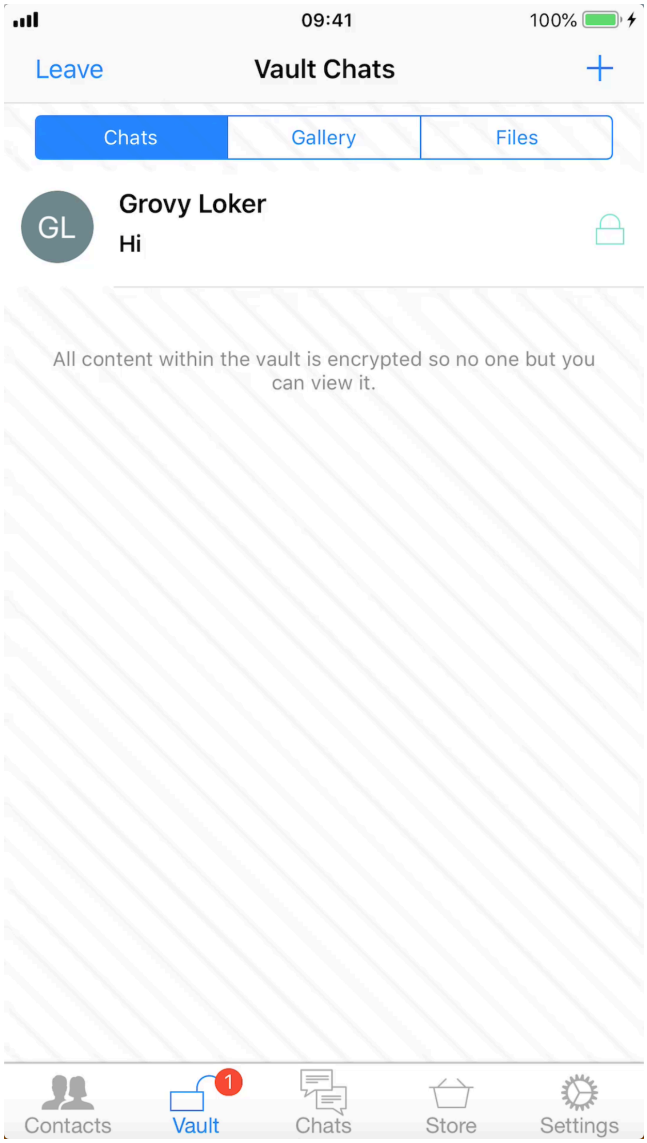


Рисунок 3.5 – Відкрите сховище даних

Щоб прочитати повідомлення і відкрити екран чату, достатньо натиснути на комірку.

Екран переписки складається з історії повідомлень між користувачами. Також серед повідомлень присутня інформація про канал передачі даних. Він може мати декілька станів. Користувач в любий момент може закрити канал або взагалі з нього вийти (Рисунок 3.6).

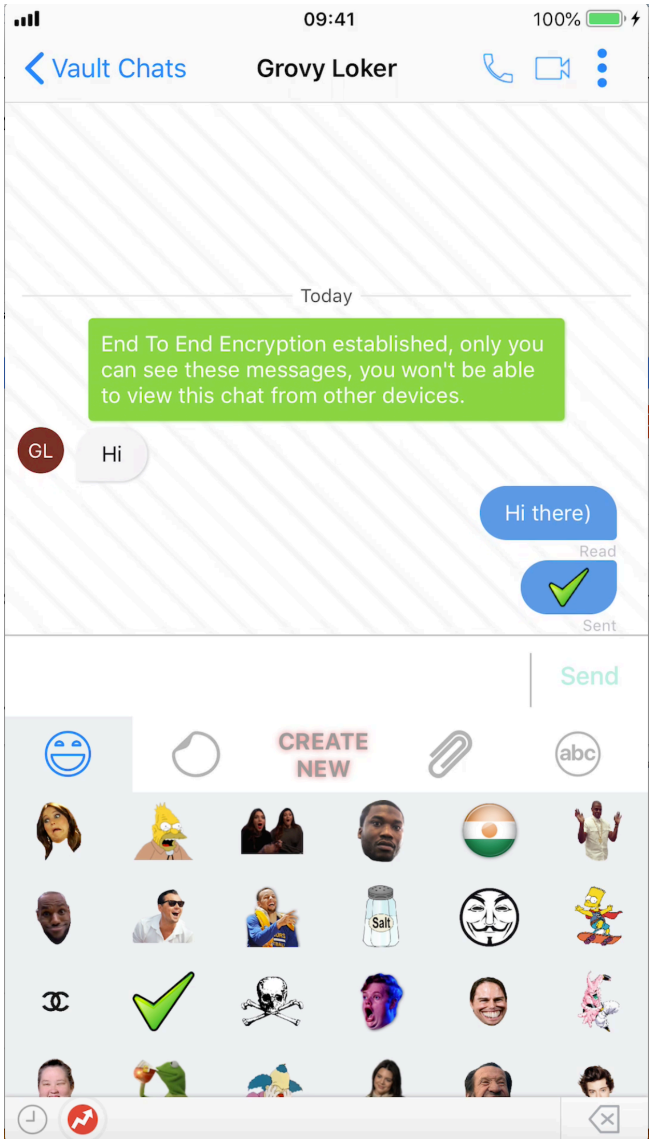


Рисунок 3.6 – Екран чату

Для того щоб відправити повідомлення, достатньо ввести його в полі внизу екрану. Також існує можливість вводити емоції. Емоції можуть бути куплені в магазині (Рисунок 3.7) або створені власноруч. Повідомлення також мають статуси які позначені під текстом повідомлення (Рисунок 3.6).

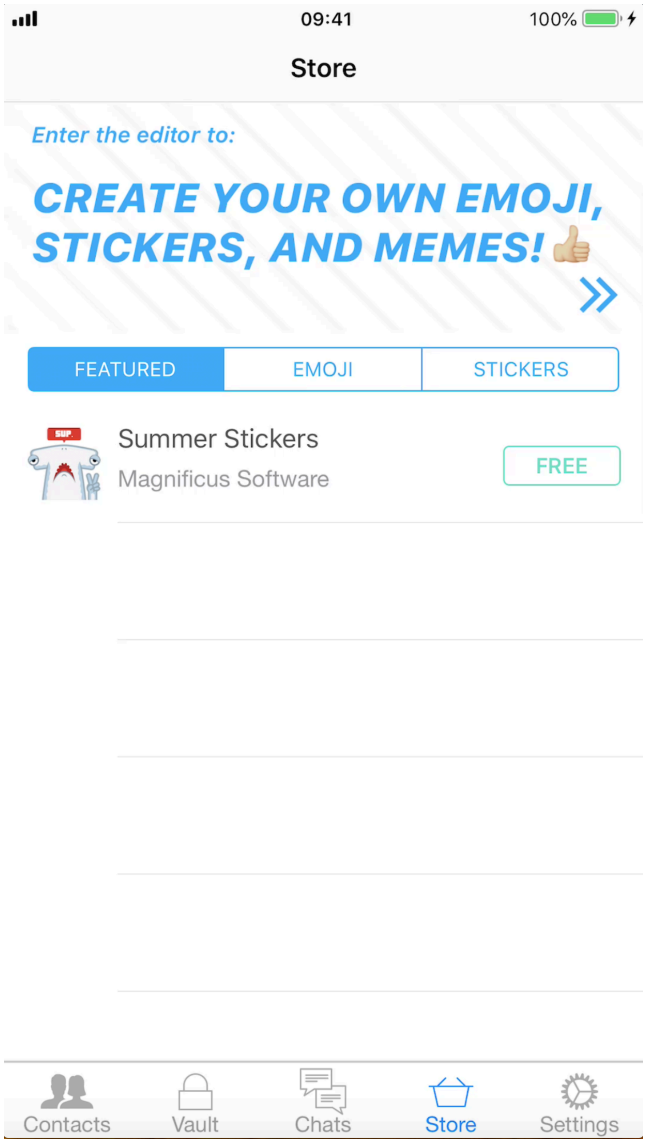


Рисунок 3.7 – Екран магазину емоцій

Останній пункт меню відповідає за налаштування застосунку (Рисунок 3.8). На цьому екрані присутня інформація про користувача з можливістю редагування її.

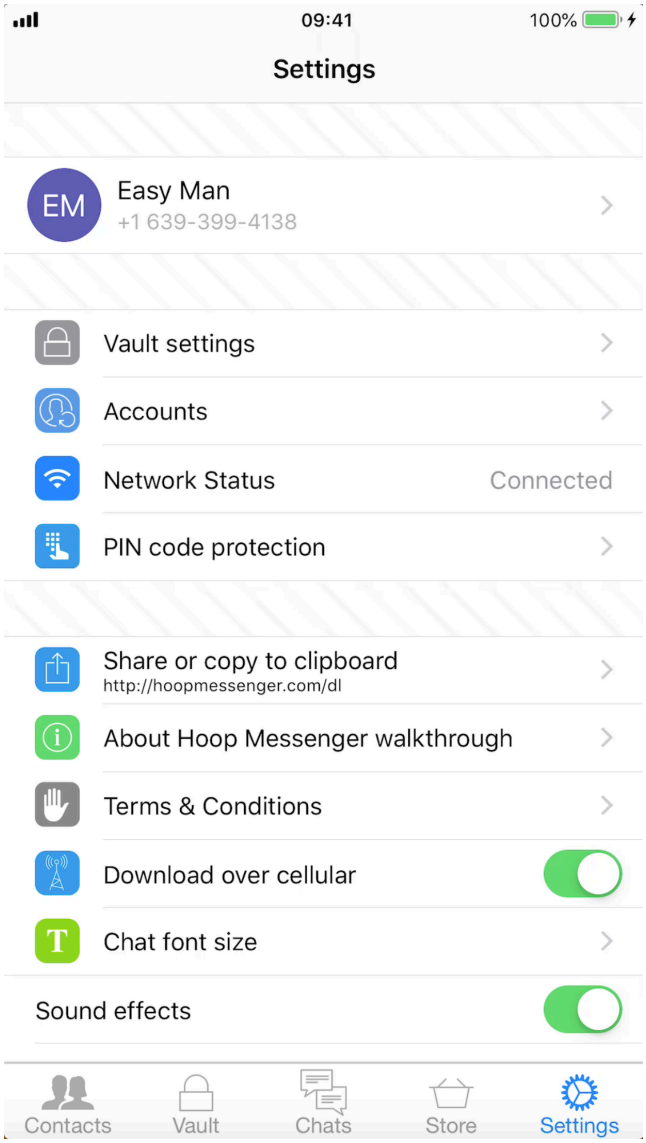


Рисунок 3.8 – Екран налаштувань

Після інформації про користувача розташовані налаштування сховища. Налаштування сховища дозволяють керувати паролем, а також містять функції синхронізації сховища та розподілення між пристроями.

На екрані налаштувань також присутня кастомізація застосунку, налаштування мережі, встановлення пін-коду при вході, інформація про застосунок та ін.

3.4 Опис технічного забезпечення

Система складається з трьох основних модулів: програмний клієнт, пристрій-сховище та сервер-трансфер (Рисунок 3.7).

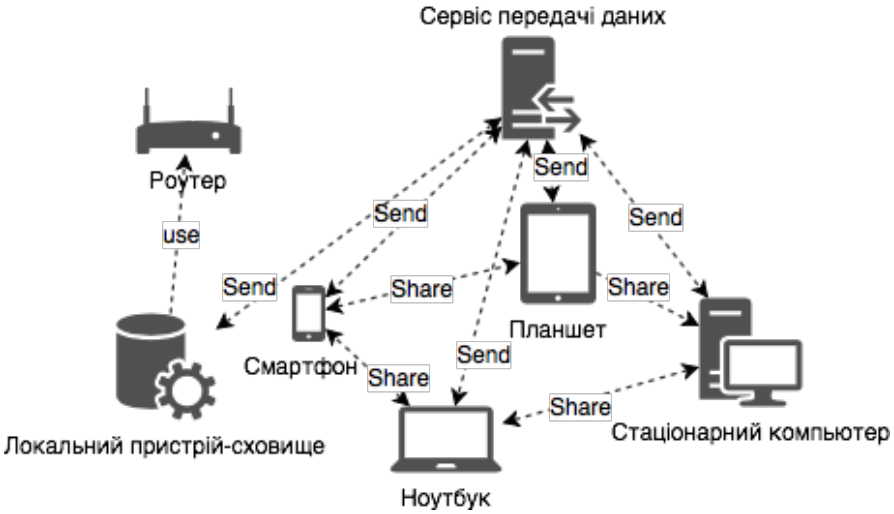


Рисунок 3.7 – Загальна діаграма системи

Програмний клієнт являє собою застосунки на платформах iOS, Android, macOS та Windows. Доступ до системи та її налаштування здійснюється тільки через клієнти. Клієнти між собою мають можливість розподіляти дані в межах одного аккаунту.

Трансферний сервіс виступає сервером для надання і керування сесіями між програмними клієнтами. Сервер займається обміном даних між користувачами, а також розподіленням та передачею даних між пристроями в межах одного аккаунту.

Локальний пристрій сховища відповідає за збереження, відправку або прийомом даних. Пристрій складається з плати та накопичувача. В плату вбудовані модулі взаємодії з мережею: Wi-Fi та Ethernet.

На рисунку 3.8 зображена діаграма розгортання системи. На ній представлені три основні модулі взаємодії системи та додатковий сервер бази даних з інформацією для авторизації користувачів.

Сервіс передачі даних має два типу інтерфейсів взаємодії. Для обміну та передачі даних між пристроями використовується Socket інтерфейс. Для авторизації використовується HTTPS з’єднання з REST запитами. Сервер передачі даних виконує функцію прошарку для взаємодії між пристроями.

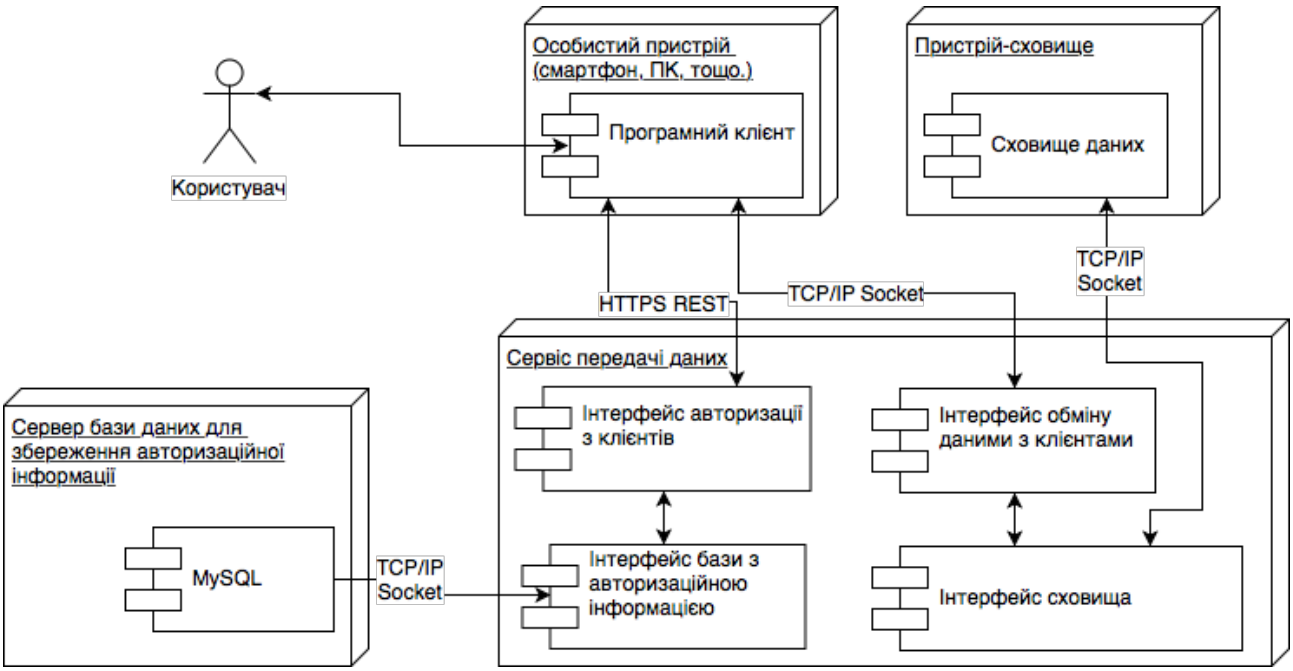


Рисунок 3.8 – Діаграма розгортання системи

Висновки до розділу

В даному розділі було розглянуто рішення з програмного та технічного забезпечення, архітектури застосунку. Були наведені діаграми послідовності та описано специфікації функцій.

Архітектура застосунку побудована на основі Viper. Цей шаблон найбільш підходить для реалізації мобільних застосунків. Нажаль сама технологія програмування під iOS– UIKit не дозволяє в повній мірі реалізувати чистий VIPER, тому шаблон реалізован на скільки це можливо

Модулі які відповідають за відправку повідомлень та трансферу відокремлені від основної системи та архітектурою теж. Вирішено в цих модулях застосувати технологію ReactiveCocoa – це фреймворк який реалізує реактивне програмування, та дає перевагу в часі, бо немає ніякої стадії очікування, модулі реагують миттєво на дії. Для цілого застосунку застосування реактивного програмування неможливе, бо тоді повністю відходить принципи SOLID.

Був наведений і обґрунтований вибір засобів розробки. Надана інструкція користувача з скріншотами і описами взаємодії з застосунком. Описане технічне забезпечення застосунку та зображена діаграма розгортання системи.

4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Опис основних ідей проекту наведений в таблиці 4.1

Таблиця 4.1 – Основні ідеї проекту

Зміст ідеї	*Напрямки застосування*	*Вигоди для користувача*
Створення системи, яка призначена для централізованого захищеного збереження даних	1. Використання серверу як трансфера, збереження даних тільки на локальних пристроях.	Повний контроль я управління своїми даними
	2. Впровадження фізичних пристроїв-сховищ.	Можливість зберігати дані на власному сервері та мати фізичний контроль над ним.
	3. Впровадження централізованої системи в бізнес	Контроль над даними в власній мережі
Розробка комплексного алгоритму шифрування для захисту каналу передачі даних	1. Використання в чаті між користувачами	Забезпечення надійного та швидкого обміну повідомленнями з іншим користувачем
	2. Використання при синхронізації даних між кінцевим пристроєм і пристроєм-сховищем	Забезпечення надійного каналу передачі даних на пристрій-сховище

Проведемо аналіз потенційних техніко-економічних переваг ідеї. Результати зображені у таблиці 4.2.

Таблиця 4.2 – Аналіз потенційних техніко-економічних переваг

No n/n	*Техніко-економічні характеристики ідеї*	*(потенційні) товари/концепції конкурентів*				*W (слабка сторона)*	*N (нейтральна сторона)*	*S (сильна сторона)*
		Мій проект	Telegram	Viber	Wahtsapp			
1.	Надійний алгоритм відправки повідомлень	+	+-		+			+
2.	Надійність збереження особистої інформації	+						+
3.	Інтерфейс	+-	+				+	
4	Гарантована відправка повідомлень		+	+	+	+		
5	Частина компоненту в системі контролю персональних даних	+						+

Аналізуючи конкурентів можна зробити висновок, що підсистема є конкуренто-спроможною. За рахунок повного контролю персональних даних користувачем та наявність повної системи доступу до даних.

Нажаль проаналізувавши ринок, конкурентів на розробку системи централізованого захищеного збереження даних не знайшлось.

4.2 Технологічний аудит ідеї проекту

Проведемо технологічний аудит ідеї проекту (Таблиця 4.3).

Таблиця 4.3 – Технологічний аудит ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1	Впровадження фізичних пристроїв-сховищ	Виготовлення сховища на базі hard disk	Наявна технологія, доступна на ринку	Доступна авторам проекту
		Виготовлення сховища на базі SSD	Наявна технологія, доступна на ринку	Доступна авторам проекту
		Виготовлення плати комунікації з сервером через інтернет	Технологія не доступна на ринку. Доступні окремі компоненти.	Не доступна авторам проекту, вимагає розробки
2	Впровадження власного трансферного серверу	Розробка серверної частини взаємодії з системою на мові Python	Наявна технологія	Доступна на ринку
		Розробка серверної частини взаємодії з системою на мові C#	Наявна технологія	Доступна на ринку

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
		Розробка кінцевого клієнту взаємодії для платформ macOS та Windows за рахунок кросплатформеного підходу	Наявна технологія	Не відома авторам
		Розробка кінцевого клієнту взаємодії для платформ macOS та Windows за рахунок рідних рішень	Наявна технологія	Доступна на ринку
3	Клієнт для контролю і взаємодії з персональною інформацією для кінцевого користувача	Розробка кінцевого клієнту взаємодії для платформ macOS та Windows за рахунок кросплатформеного підходу	Наявна технологія	Не відома авторам
		Розробка кінцевого клієнту взаємодії для платформ macOS та Windows за рахунок рідних рішень	Наявна технологія	Доступна на ринку

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
4		Розробка кінцевого клієнту взаємодії для iOS та Android за рахунок кросплатформеного підходу	Наявна технологія	Не відома авторам
5		Розробка кінцевого клієнту взаємодії для iOS та Android за рахунок рідних рішень	Наявна технологія	Доступна на ринку

Проаналізувавши технології можна зробити висновок, що для реалізації та впровадження фізичних пристроїв-сховищ буде доцільнішим використовувати hard disk, бо вона має більший розмір даних і меншу коштовність. Плата для комунікації з сервером на ринку відсутня, тому вимагає розробки.

Що стосується клієнтів для кінцевих користувачів та бізнесу, краще використовувати рідні рішення, бо вони відомі авторам проекту та мають більшу якість порівняно з кроссплатформеними рішеннями.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Виведемо попередню характеристику потенційного ринку стартап-проекту (Таблиця 4.4).

Таблиця 4.4 – Характеристика потенційного ринку стартап-проекту

№	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	1 000 000+
3	Динаміка ринку (якісна оцінка)	Зростає

№	*Показники стану ринку (найменування)*	*Характеристика*
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Перевірка захищеності даних по стандартах
6	Середня норма рентабельності в галузі (або по ринку), %	85%

Ринок є привабливим для входження і має перспективу за рахунок відсутності обмежень та динаміки ринку.

Для підвищення довіри ринку та реклами необхідно пройти сертифікацію якості захищеності системи.

Розглянемо потенціальні групи клієнтів у вигляді таблиці 4.5:

Таблиця 4.5 – Потенціальні групи клієнтів

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1	Захищений обмін повідомленнями	Кінцевий користувач	Безкоштовна експлуатація Підтримка максимальної кількості платформ	Зручний у використанні Захищений канал відправки
2	Захищене збереження персональних даних	Кінцевий користувач	Використання якомога меншої кількості пам’яті. Швидкість шифрування та дешифрування	Прозорий доступ до даних Наявність сертифікатів захищеності

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
3	Контроль та управління даними	Кінцевий користувач	Простий доступ з любої платформи до даних Контроль даних в любий момент експлуатації	Повний контроль до файлів і даних
4	Можливість фізичного доступу до даних	Кінцевий користувач	Простота у технічному використанні. Надання різних типів пристроїв в залежності від ціни	Зручність у використанні пристроїв-сховищ Надійність доступу в любий момент
5	Можливість впровадження системи в бізнес з повним контролем трансферу і доступу до даних	Бізнес корпорації	Можливість розгортання на простих машинах. Варіативність ціни підписок	Простота розгортання Наявність сертифікатів якості захисту Зручність використанні

Після аналізу цільової аудиторії, можна зробити висновок, що товар буде заходити на два різних ринку: ринок бізнес впровадження та ринок кінцевих користувачів.

Проведемо аналіз ринкового середовища, складемо таблиці факторів загроз та факторів можливостей (Таблиці 4.6, 4.7)

Таблиця 4.6 – Таблиця факторів загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Ціна за пристрою-сховища	Велика ціна за пристрій-сховище	Перехід на більш дешеве виготовлення деяких компонентів, можливість впровадження акціонних підписок
2.	Складність використання	Складність використанні і розуміння системи	Підвищення рівню пояснень, навчань системи та реклами.

Таблиця 4.7 – Таблиця факторів можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання попиту	Можливість захоплення більшого ринку	Купівля більшої кількості ресурсів на виготовлення продукту
2	Зміна пріоритетів користувачів	Зміна пріоритетів користувачів на збільшення використання відправки повідомлень	Виділення ресурсів та удосконалення методів відправки і захисту повідомлень

Згідно таблиць, можна зробити висновок що можливе утворення деяких факторів зміни моделі розробки і впровадження продукту.

Проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку (Таблиця 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Тип конкуренції - монополістична	Існують декілька компаній які пропонують збереження та передачу даних але не ідентичні даній	Відмінність та конкурентоспроможність компанії полягає в збереженні всієї інформації локально на пристроях. Надання пристроїв-сховищ для персонального користування.
2. Рівень конкурентної боротьби - національний	Компанії конкуренти представляють різні країни	Розробка багатомовного сервісу підтримки користувачів та інструкцій ознайомлення і купівлі продукту.
3. Галузева ознака - міжгалузева	Компанія конкурує в різних галузях: обмін персональними даними, збереження персональних даних та продаж і використання пристроїв-сховищ.	Використання різних галузей як одного продукту підвищує конкурентоспроможність компанії.
4. Конкуренція за видами товарів - товарно-родова	Компанія в більшості конкурує з схожими функціями, але за	Новий підхід впровадження пристроїв-сховищ, підвищить

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства*
	різними типами продуктів.	зацікавленість і надійність підходу.
5. Характер конкурентних переваг - нецінова	Ціна відходить на другий план конкурентоспроможності	Основний наголос іде на якість і тип продукту, але ціна також ринкова.
6. За інтенсивністю - марочна	Позначення продукту маркою компанії.	Так як продукт реалізується за різними типами, марочна політика забезпечить взнаваємось на ринку і довіру користувачів.

Після аналізу конкуренції проведем більш детальний аналіз умов конкуренції в галузі (Таблиця 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари-замінники*
Складові аналізу	Telegram, Viber, Wahtsapp	Компанії з новим підходом збереження та передачі даних.	Програмні платформи з великою кількістю користувачів (Apple, Google, Microsoft)	Потреба захисту персональних даних. Більший контроль над власними даними.	Можливість ручного збереження та контролю даних на примітивних пристроях

	Прямі конкуренти в галузі	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари-замінники*
Висновки:	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	Існує можливість входу на ринок з більш прогресивними рішеннями. Існують потенційні конкуренти, строк виходу залежить від складності підходу та зацікавленості користувачами	Відповідність заявленим критеріям (Software markets). Точність розрахунків проектування (пристрої-сховища)	Клієнти потребують швидкої взаємодії і опрацювання паралельно з надійністю	Обмеження для роботи на ринку через товари замінники мало імовірні або відсутні взагалі.

З огляду на конкурентну ситуацію, можна зробити висновок, що для можливої роботи на ринку проект має містити нові підходи до надання функціональності, якість продукту має бути не нижча за аналоги.

На основі аналізу конкуренції, проведеного в таблиці 4.9, а також із урахуванням характеристик ідеї проекту (таблиця 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (таблиці 4.6-4.7) визначаємо та обґрунтовуємо перелік факторів конкурентоспроможності. Аналіз оформлюємо за таблицею 10

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
1	Можливість підключення пристроїв-сховищ	Відсутня конкуренція в даній функції серед компаній. Цей фактор Дасть змогу користувачам контролювати фізичне розташування даних
2	Можливість використання власного трансферного серверу	Відсутня конкуренція в даній функції серед компаній. Дасть змогу компаніям, з власною системою передачі інформації, захищати потік даних між користувачами.
3	Реалізація клієнту для контролю і взаємодії з персональною інформацією, для кінцевого користувача, на різних платформах	Дасть вибір користувачам зручних платформ для контролю і взаємодії зі персональною інформацією.
4	Створення власної персональної системи контролю та передачі даних	Об’єднання різних платформ з фізичними пристроями дасть змогу користувачу контролювати та передавати персональну інформацію в одній системі.

За визначеними факторами конкурентоспроможності (Таблиця 4.10) проведемо аналіз сильних та слабких сторін стартап-проекту (Таблиця 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін системи

№ п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з системою*						
			—- 3	*—- 2*	*—- 1*	*0 0*	*+ 1*	*+ 2*	*+ 3*
1	Можливість підключення пристроїв-сховищ	18				+			
2	Можливість використання власного трансферного серверу	10				+			
3	Реалізація клієнту для контролю і взаємодії з персональною інформацією, для кінцевого користувача, на різних платформах	15		+					
4	Створення власної персональної системи контролю та передачі даних	12			+				

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості

цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза). Виведемо SWOT- аналіз стартап-проекту (Таблиця 4.12).

Таблиця 4.12 – SWOT- аналіз стартап-проекту

Сильні сторони: Можливість підключення пристроїв-сховищ Можливість використання власного трансферного серверу Реалізація клієнту взаємодії на різних платформах Власна персональна система контролю та передачі даних	Слабкі сторони: Можливо складний та незрозумілий інтерфейс Фізичний злам пристроя-сховища
Можливості: Зростання попиту Зміна приорітетів користувачів в середені системи	Загрози: Ціна за пристрої-сховища Складність використання

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок

Визначені альтернативи аналізуємо з точки зору строків та ймовірності отримання ресурсів (Таблиця 4.13).

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1	Вихід на ринок з клієнтів з локальним збереженням даних, на популярних платформах. Далі підключення пристроїв	95%	6-8 місяців

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
	сховищ. Запровадження серверу обміну та передачі даних.		
2	Вихід на ринок з клієнтів на всіх платформах. Наступні кроки з пункту 1.	85%	12-18 місяців
3	Вихід на ринок з популярних клієнтів та пристроїв сховищ. Далі запровадження серверу обміну та передачі даних.	65%	10-12 місяців
4	Вихід на ринок з популярних клієнтів, пристроїв сховищ та серверу обміну та передачі даних.	50%	18-24 місяця

З таблиці 4.13 можна зробити висновок, що доцільно обрати альтернативу 1. Альтернатива 1 має найбільшу ймовірність отримання ресурсів бо залежить тільки від реалізації програмного забезпечення, на існуючих платформах. Строки реалізації альтернативи 1 – найменші. Альтернатива 1 дозволить запустити продукт в реалізацію і після займатись розробкою та рекламою пристроїв-сховищ. Вихід пристроїв сховищ буде більш очікуваним та буде орієнтовна кількість замовлень на перші партії. Так як запровадження трансферного сервісу орієнтується на бізнес і неможливий без клієнтів та пристроїв, цей процес відходить на пізнішу реалізацію.

4.4 Розроблення ринкової стратегії проекту

Визначмо стратегії охоплення ринку: опис цільових груп потенційних споживачів. Виберемо цільові групи споживачів таблиця 4.14.

Таблиця 4.14 – Вибір цільових груп споживачів

No n/n	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1	Бізнес компанії	Висока	Високий	Низька	Висока
2	Школярі	Низька	Середній	Низька	Висока
3	Студенти	Висока	Середній	Середня	Висока
4	Робітники в сфері медіа	Висока	Високий	Висока	Середня
5	Туристи	Висока	Середній	Середня	Висока
6	Користувачі з активним оновленням і трансфером даних	Висока	Високий	Висока	Середня
Які цільові групи обрано: Робітники в сфері медіа, Користувачі з активним оновленням і трансфером даних, Туристи, Бізнес компанії					

Після аналізу обрані цільові групи які поділяються на дві категорії: бізнес та кінцевий користувач. Для кінцевого користувача цільові групи які пов’язані з збереженням та обміном середніх або великих об’ємів даних. Стратегія маркетингу визначена диференційовано-масова.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (Таблиця 4.15)

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку**
1	Вихід на ринок з клієнтів з локальним збереженням даних, на популярних платформах. Далі підключення пристроїв сховищ.	Впровадження безкоштовного користування системою, отримання заробітку на другій фазі з пристроїв-сховищ. Заробіток та продажах трансферних систем для бізнесу.	Забезпечення безкоштовного користування системами. Низька ціна на пристрої сховища. Заробіток на об’ємах продажів.	Стратегія лідерства по витратах
2	Запровадження тренсферного серверу.	Надання ключових конкурентоспроможних позицій по черзі для збільшення попиту. Забезпечення надійного функціонування всіх модулей. Легкий вихід на бізнес ринок після успішного впровадження для	Збереження всіх даних локально на пристроях. Використання пристроїв сховищ. Трансфер системи для впровадження в бізнес	Стратегія диференціації

		кінцевого користувача.		
№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку**
3		Так як більший попит планується від кінцевих користувачів, відповідно до альтернативи, реклама і заохочування буде спрямовано на користувачей зі збереженням та обміном середніх або великих об’ємів даних. Далі планується впровадження у бізнес.	Введення продукту у сферах з обробкою та трансфером великих об’ємів даних. Можливість впровадження у бізнес.	Стратегія спеціалізації

Після аналізу отримуємо, що стратегія лідерства по витратах може погано вплинути на якість продукту. Стратегія спеціалізації не може бути обрана, бо система має забезпечувати потреби різних цільових груп. Тому обрана стратегія диференціації. Основна ідея проекту в забезпеченні надійного функціонування системи з високою якістю продуктів.

Виберімо стратегії конкурентної поведінки (таблиця 4.16)

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	*Чи є проект «першопрохідцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки**
1	Так	Планується забирати існуючих користувачів з інших систем, а також заохочувати нових.	Основні характеристики товару не копіюються. Буде схожі типи обміну даними, наприклад повідомлення, передача медіа файлів	Стратегія лідера

Відповідно до аналізу, була обрана стратегія лідера, бо система є першопрохідцем на ринку і, насамперед планується забезпечити лідерство.

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування, що полягає у формуванні ринкової позиції за яким споживачі мають ідентифікувати торгівельну марку/проект (Таблиця 4.17).

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
1	Зручний обмін даними та захищений канал обміну	Стратегія диференціації	Розробка та аналіз власних алгоритмів захисту. Надання зручного інтерфейсу взаємодії.	Швидкість відправки даних. Контроль даних. Простий в експлуатації інтерфейс.
2	Прозорий доступ до даних Наявність сертифікатів захищеності		Збереження даних локально. Зручний інтерфейс контролю даних. Проходження аудиту безпеки.	Наявність сертифікатів захищеності. Простий в експлуатації інтерфейс контролю даних. Локальне збереження всіх даних та розподілення їх.
3	Повний контроль до файлів і даних		Збереження на власних пристроях а також на пристроях - сховищах	Використання пристроїв-сховищ.
4	Зручність у використанні пристроїв-сховищ		Надання зручного інтерфейсу взаємодії та підключення пристроїв-сховищ.	Зручність та мобільність використання пристроїв-сховищ

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
5	Простота розгортання та зручність використання бізнесом		Можливість впровадження системи в бізнес. Надання інструкцій щодо розгортання та підтримки клієнтів.	Можливість впровадження в бізнес та підтримка клієнтів

Після аналізу можна зробити висновок, що були обрані стратегії диференціації та лідера. Це означає що основний напрямок розвитку це поступове введення компонентів системи, базуючись на якості продукту та новизні.

4.5 Розроблення маркетингової програми стартап-проекту

Сформуємо маркетингову концепцію товару, яку отримає споживач. Для цього підсумуємо результати попереднього аналізу конкурентоспроможності товару (Таблиця 4.18).

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
	Захищений обмін повідомленнями	Сертифікована захищеність	Захищеність каналу власними алгоритмами шифрування.
	Захищене збереження персональних даних	Отримання фізичного доступу до даних.	Збереження даних на локальних носіях.
	Контроль та управління даними	Доступ до даних з всіх популярних платформ.	Реліз продукту на багатьох платформах. Синхронізація між платіормами
	Можливість фізичного доступу до даних	Використання пристроїв-сховищ. Збереження на особистих пристроях	Розподілена система збереження даних. Підримка та трансфер даних між пристроями користувача.
	Можливість впровадження системи в бізнес з повним контролем трансферу і доступу до даних	Можливість керування системою як підсистемою. Гнучкість впровадження, в залежності від пристроїв	Впровадження сервісу обміну та передачі даних

Розробимо трирівневу маркетингова модель товару: уточнимо ідею продукту, його фізичні складові, особливості процесу його надання (Таблиця 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	*Сутність та складові*		
I. Товар за задумом	Можливість захищеного збереження та передачі даних. Фізичний та віддалений доступ до даних.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Персональна система контролю та передачі даних на різних платформах	Нм	Тх
	2. Захищене збереження даних	Нм	Тх
	3. Захищений канал передачі даних	М	Тл
	4. Використання власного пристрою-сховища, для фізичного доступу до даних.	Нм	Тх
	5. Можливість використання власного трансферного сервісу		
	Якість: тестування пристрою-сховища. Наявність Сертифікату захищеності.		
	Програмна частина буде виходити в реліз через платформні магазини. Пристрої-сховища будуть мати вигляд приладу з SSD та платою керування.		
	Марка: Sendstore (поєднання відправки і збереження даних)		
III. Товар із підкріпленням	До продажу: можливість використання безплатних функцій (лімітована відправка та збереження даних)		
	Після продажу: інструкція з встановлення пристрою-сховища, інструкція використання платформи. Підтримка клієнтів.		

За рахунок чого потенційний товар буде захищено від копіювання: патента на розробку і впровадження пристроїв-сховищ. Патент на використання алгоритму захисту каналу передачі даних. Патент на розподілену систему локального збереження даних, використання серверу як трансферного та впровадження його в інші системи.

Визначмо цінові межі, якими треба керуватись, при встановлені ціни на потенційний товар. Товарів-замінників не існує, тому будемо розглядати товари аналоги (Таблиця 4.20).

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	*Назва аналогу*	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
Збереження та трансфер даних (розширена версія) – підписка в місяць				
1	Google cloud	2.28$/100 Gb	700-2000$	2-3$
2	iCloud	2$/100Gb		
3	Drop box	1$/100gb		
Пристрій-сховище				
1	Переносний HDD	80$/1Tb	700-2000$	120-150$
2	Зовнішній носій для модему	120$/1Tb		
Система для впровадження в бізнес – підписка в місяць				
1	Atlassian	7000$/ 100 корист.	500 000-700 000$	8 000-10 000$

Визначмо оптимальні системи збуту (Таблиця 4.21)

Таблиця 4.21 Формування системи збуту

№ п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
1	Купівля програмного забезпечення через платформні магазини	Надання можливості завантаження та встановлення програмних клієнтів	одноріневий	Оптимальна система збуту полягає в релізі програмних клієнтів на платформні магазини і дистрибуції через них до кінцевого користувача
2	Купівля пристроїв-сховищ	Надання можливості купівлі в магазинах та онлайн	Нульовий(з сайту системи) та дворівневий (від третьої сторони)	Оптимальна система збуту полягає в купівлі пристроїв як з сайту системи так і в зручному для користувача магазині
3	Купівля трансферної системи бізнесом	Надання можливості встановлення ті підключення системи	Нульовий	Оптимальна система полягає в замовленні, встановленні та підключенні системи напряму.

Розробимо концепцію маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (Таблиця 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій, якими користуються цільові клієнти*	*Ключові позиції, обрані для позиціонування*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*
1	Потреба у захищеному збереженні та передачі даних	Соціальні мережі, інтернет магазини, сайти новин, дороги, місця масового скупчення людей,	Захищений канал передачі даних. Дані не зберігаються на сервері. Всі дані знаходяться у користувача.	Донести до користувача новий спосіб збереження даних в розподіленій локальній системі на персональних пристроях. Передача даних у захищеному каналі	Ніхто не має фізично даних крім користувача. Всі дані знаходяться в особистому фізичному доступі, а канал передачі має високий захист.
2	Збільшення розміру локального сховища		Пристрій-сховище для збереження даних, який передає дані	Донести до користувача можливість збереження і передачі всіх	Всі ваші дані у вас вдома або де ви забажаєте. Повний

№ п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій, якими користуються цільові клієнти*	*Ключові позиції, обрані для позиціонування*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*
			через модулі зв’язку	даних з власного пристрою сховища, до якого є фізичний доступ.	фізичний контроль над даними. Можливість розвантаження пам’яті персональних пристроїв і перенесення даних на пристрій-сховище
3	Потреба у власному трансферному сервісі бізнесом		Впровадження системи для бізнесу	Донести бізнесу безпечне використання і контроль всіх даних в системах	Використовуй розподілену систему збереження даних у власній системі. Контролюй трансфер даних в системі.

В процесі аналізу була розроблена маркетингова програма, що включає в себе концепції потенційного товару, три рівні моделі товару. Були визначені межі встановлення цін на різні типи товарів. Обрані системи збуту в залежності від типу товару. Сформована концепція маркетингових комунікацій.

Висновки до розділу

Під час аналізу були описані ідеї проекту, проведений технічний аудит ідеї проекту. Проведений аналіз ринкових можливостей запуску стартап-проекту. Розроблена ринкова стратегія та маркетингова програма проекту.

Можна зазначити що ринкова комерціалізація проекту можлива, бо існує попит на такий тип послуг з рентабельністю на ринку. Перспектива продукту дуже висока, бо не існує на даний момент конкурентів і бар'єрів входження на ринок. Існує цільова аудиторія з високим рівнем платіжоздатності.

Доцільно обрати альтернативу впровадження з виходом на ринок з програмних-клієнтів з локальним збереженням даних, на популярних платформах. Далі підключення пристроїв сховищ. Запровадження серверу передачі і обміну даних.

Після реалізації на ринок, система має майбутнє не тільки з підтримкою існуючих клієнтів, а і з подальшою імплементацією та удосконаленням.

ВИСНОВКИ

В цій дисертації була поставлена мета підвищення ступеню захищеності персональних даних та надання користувачу повного контролю над ними. Ця мета була досягнена за рахунок виконання поставлених задач. Була доведена актуальність і необхідність розробки даної системи.

В розділі проектних рішень з розробки інформаційних систем, було описане предметне середовище та бізнес процеси системи. Розкриті цілі та завдання проектування та розробки системи. Розглянута повна система та її складові. Описана доцільність розгортання підсистеми взаємодії з користувачем у вигляді мобільного застосунку. Наведені діаграми діяльності та варіантів використання мобільного застосунку. Був проведений аналіз вхідних та вихідних даних, наведена структура бази даних. Обрані формати передачі даних в захищеному каналі.

В розділі моделей та методів розробки була поставлена і вирішена задача розробки комплексного асиметричного алгоритму шифрування з динамічним ключем. Вирішення задачі досягалось шляхом створення комплексного алгоритму на базі гібридної криптосистеми з додаванням динамічного ключа. Гібридна система базується на використанні різних типів алгоритмів, зокрема симетричного та асиметричного. У розділі були проаналізовані існуючі алгоритми та обрані підходящі для гібридного впровадження.

Був розроблений алгоритм зв’язку поточних блоків з попередніми ітераціями, тобто динамічний ключ в комплексному підході. Особливість цього ключа полягає в тому, що він може бути впроваджений, як додатковий, в любі алгоритми блокового шифрування. Складність шифрування динамічного ключа обрана одна з простих, для економії ресурсів, але алгоритм може бути покращений, бо він розширюваний.

Комплексний алгоритм був розроблений та наведений структурно. Функції шифрування та дешифрування були покроково описані та зображені.

В розділі програмного та технічного забезпечення були описані засоби розробки та обґрунтований вибір їх за характеристиками. Була зображена та описана архітектура мобільного застосунку. На діаграмах розгортання була показана повна система та взаємодії підсистем. Була наведена інструкція користувача для взаємодії з

мобільним застосунком. Зображені скріншоти основних екранів з описанням функціональності.

В розділі розробки стартап-проекту були описані та проаналізовані ідеї проекту. Були попередньо розглянуті ринкові можливості запуску проекту. Зазначена можливість комерціалізації проекту, а також виведена висока перспектива продукту.

Система має високу ймовірність успішного старту проекту та подальшого удосконалення. Чітко сформульований план виходу на ринок і подальших кроків. Засоби розробки і впровадження продукту вибрані сучасні і високоякісні. Математична складова продукту має добре проаналізовані та розроблені алгоритми захисту передачі та збереження персональної інформації. Програмне забезпечення для взаємодії з користувачем, у вигляді мобільного застосунку, розроблене з високим рівнем захисту персональної інформації, водночас зі зручним і простим інтерфейсом, що гарантує швидку адаптацію і довіру до системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Johns M. “Introducing XMPP” [Електронний ресурс] / Johns M. – IBM, 2010. – Режим доступу: https://www.ibm.com/developerworks/ru/library/x-xmppintro/index.html

2. Villanueva J, “Symmetric vs Asymmetric Encryption” [Електронний ресурс] / John Carl Villanueva – Jscape, 2015. – Режим доступу: https://www.jscape.com/blog/bid/84422/Symmetric-vs-Asymmetric-Encryption.

3. Ramesh.A. “Performance Analysis of Encryption algorithms for Information Security” / Ramesh.A, Suruliandi.A – San Jose, USA, 2013. – pp. 78-97

4. Rafael A. “Symmetric and Asymmetric Encryption” [Електронний ресурс] / Rafael A. – Medium, 2017. – Режим доступу: https://hackernoon.com/symmetric-and-asymmetric-encryption-5122f9ec65b1

5. P. Jindal “Analyzing the Security-Performance Trade-off in Block Ciphers” / P. Jindal, B. Singh – California, 2015. – pp. 326 - 331

6. Sheffer Y., “Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)” [Електронний ресурс] / Sheffer Y., Holz R., Saint-Andre P. – RFC 7525, 2015. – Режим доступу: https://www.rfceditor.org/rfc/pdfrfc/rfc7525.txt.pdf

7. J.-H. Hong “Radix-4 modular multiplication and exponentiation algorithms for the RSA public-key cryptosystem” / J.-H. Hong, C.-W. Wu. – Design Automation Conference (ASP-DAC 2000), 2017. – pp. 23-45

8. Johannes A. "Introduction to Cryptography", 2nd ed. / Johannes A. – Springer, 2016. – pp. 71-137

9. C. Kaufman, “Network Security: Private Communication in a Public World” / C. Kaufman, R. Perlman, and M. Speciner. – Prentice Hall PTR, 2015. – pp. 75-90

10. M. S. Merkow, “The Complete Guide to Internet Security” / M. S. Merkow, J. Breithaupt. – AMACOM, 2018. – pp. 34-52

11. Nadeem "A Performance Comparison of Data Encryption Algorithms" / Nadeem – 2015. – pp. 70-95

12. X. Laai. “On the Design and Security of Block Ciphers” / X. Laai – Hartung-Gorre Veerlag, 2012. – pp. 24-56

13. J. Burke “Architectural Support for Fast Symmetric-Key Cryptography” / J. Burke, J. McDonald, T. Austin. – ASPLOS, 2000. – pp. 43-80

14. J. L. Hennessy “Computer Architecture in Algorythms” / J. L. Hennessy, D. A. Patterson – Palo Alto, California, 2017. – pp. 63-76

15. Timo Bingmann, “Speedtest and Comparison of Open-Source Cryptography Libraries and Compiler Flags” [Електронний ресурс] / Timo Bingmann – 2008. – Режим доступу: https://panthema.net/2008/0714-cryptography-speedtest-comparison/

16. Phillip Rogaway, “Evaluation of Some Blockcipher Modes of Operation” / Phillip Rogaway – Лютий 10, 2017. – pp. 32-67

17. Wei Dai “Benchmarks” [Електронний ресурс] / Wei Dai – Режим доступу: https://www.cryptopp.com/benchmarks.html

18. Mushtaque A. “Evaluation of Encryption Algorithms” / Mushtaque A. – 2014. – pp. 80-90

19. Скидан Д. О. “Комплексний асиметричний алгоритм шифрування з динамічним ключем” / Скидан Д.О, Жданова О. Г. // Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2018) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 29-30 грудня 2018. – С. 139-143.

20. Скидан Д.О. “Аналіз симетричних алгоритмів шифрування для впровадження у гібридну криптосистему” / Скидан Д. О. // Актуальні наукові дослідження в сучасному світі – iSience – 2018. – С. 54-60

21. Meng To “The Power of Xcode” [Електронний ресурс] / Meng To – Режим доступу: https://medium.com/learning-xcode-as-a-designer/the-power-of-xcode-afcfd3f9128b

22. “The Objective-C Programming Language” [Електронний ресурс] // Apple – Режим доступу: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html

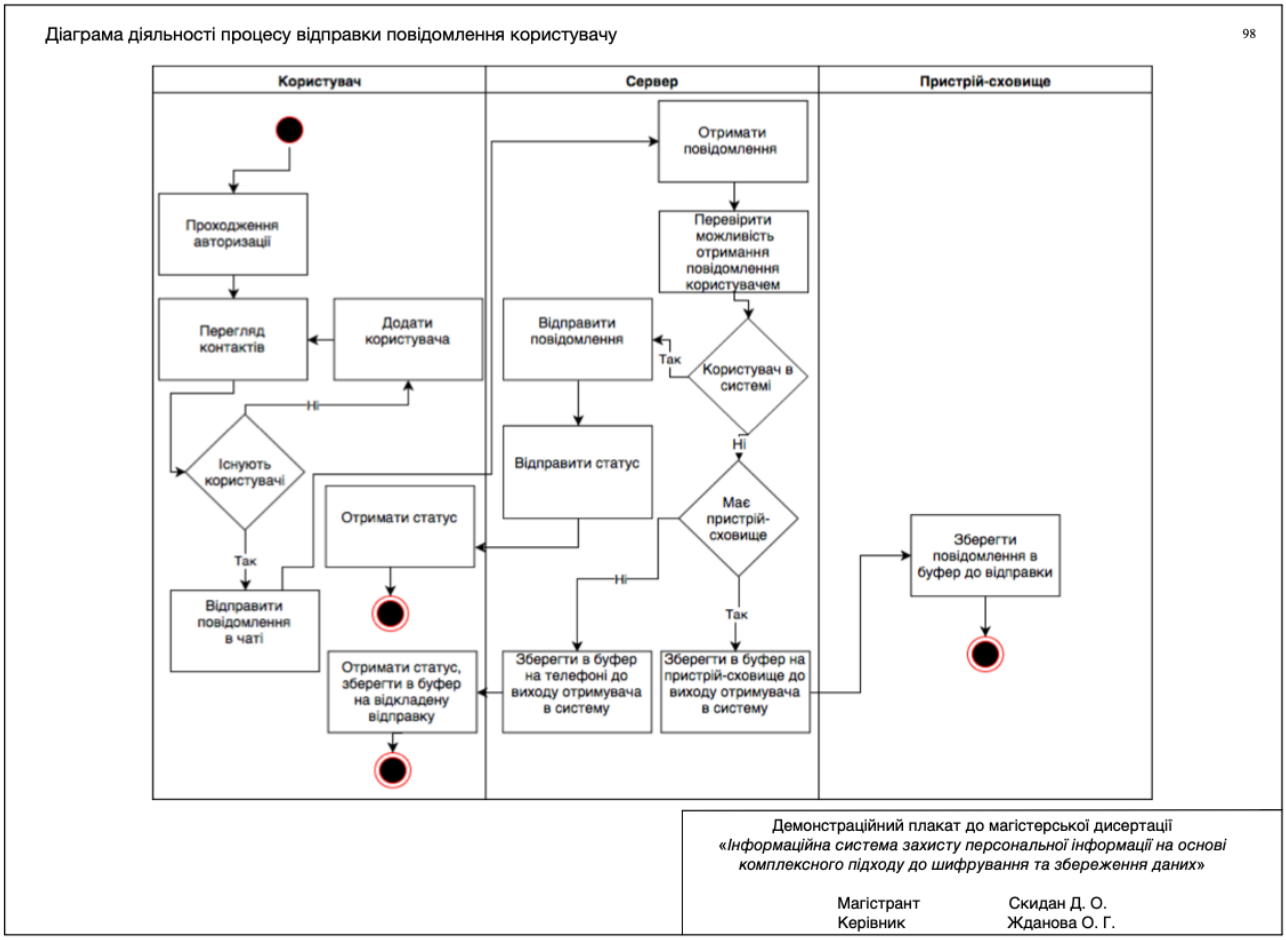
23. Daniel Eggert “Core Data Overview” [Електронний ресурс] / Daniel Eggert – Режим доступу: https://www.objc.io/issues/4-core-data/core-data-overview/

ДОДАТОК А

ГРАФІЧНИЙ МАТЕРІАЛ

Діаграма діяльності процесу відправки повідомлення користувачу

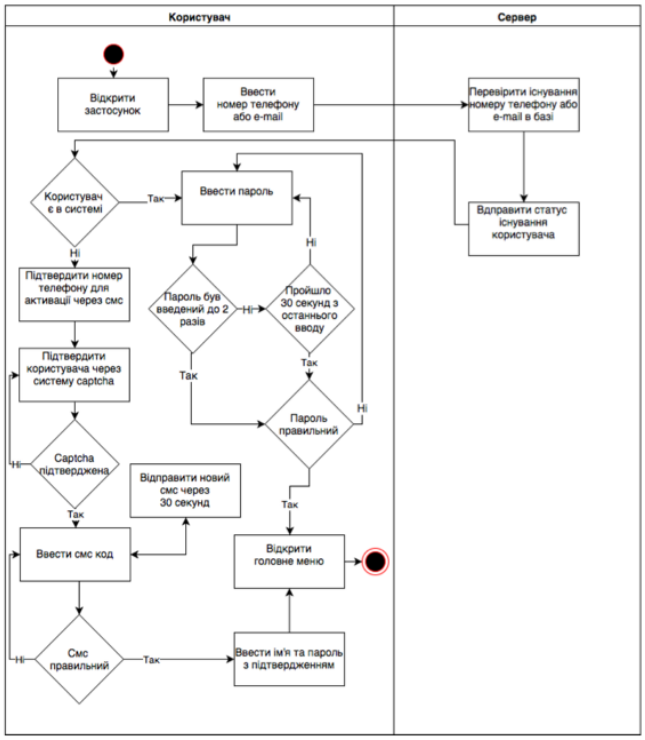
Заглушка для номеру сторінки



Діаграма діяльності авторизації в системі

Діаграма діяльності авторизації в системі

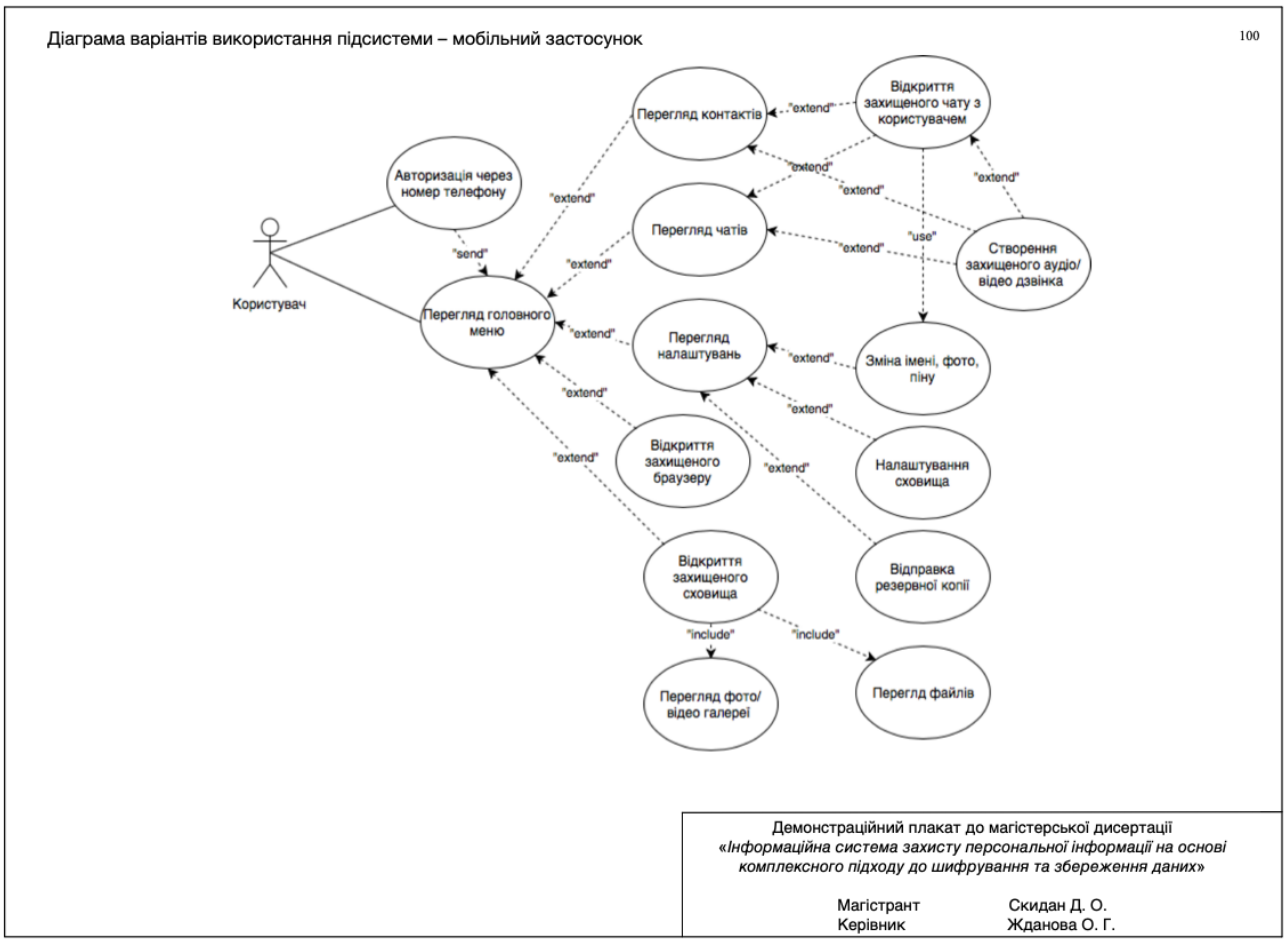
99



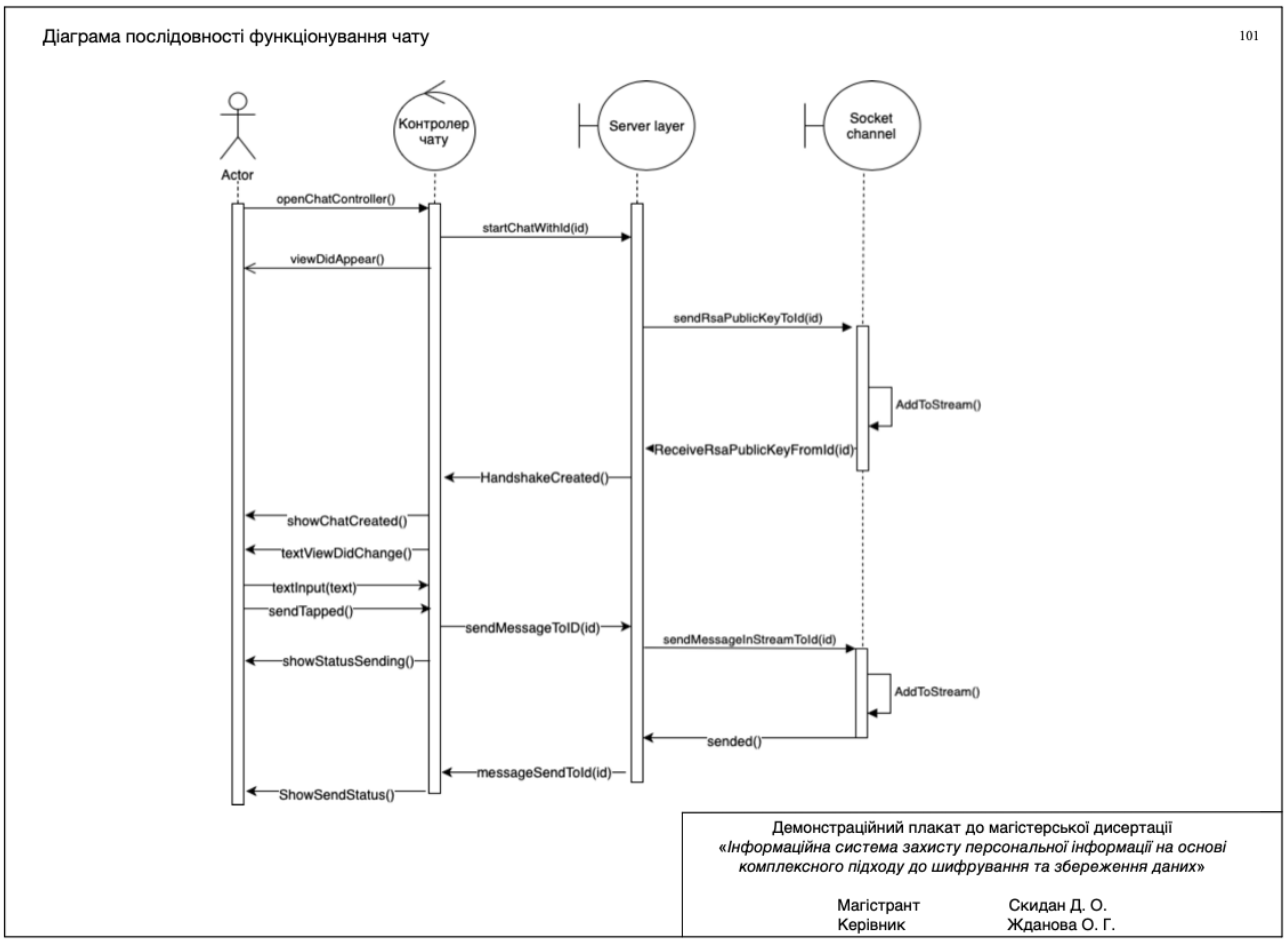
Демонстраційний плакат до магістерської дисертації
«*Інформаційна система захисту персональної інформації на основі комплексного підходу до шифрування та збереження даних*»

Магістрант Скидан Д. О.
Керівник Жданова О. Г.

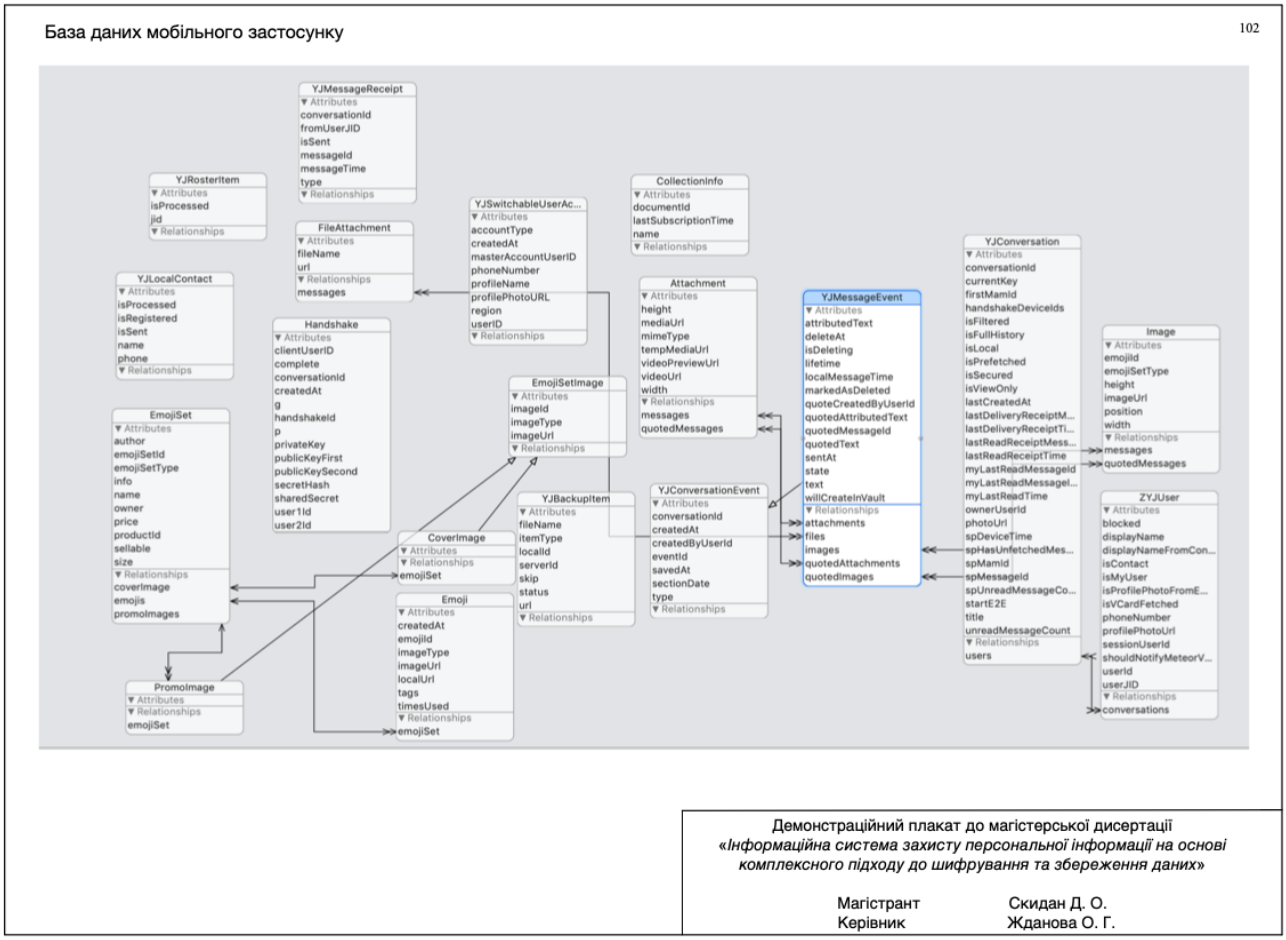
Діаграма варіантів використання підсистеми – мобільний застосунок



Діаграма послідовності функціонування чату



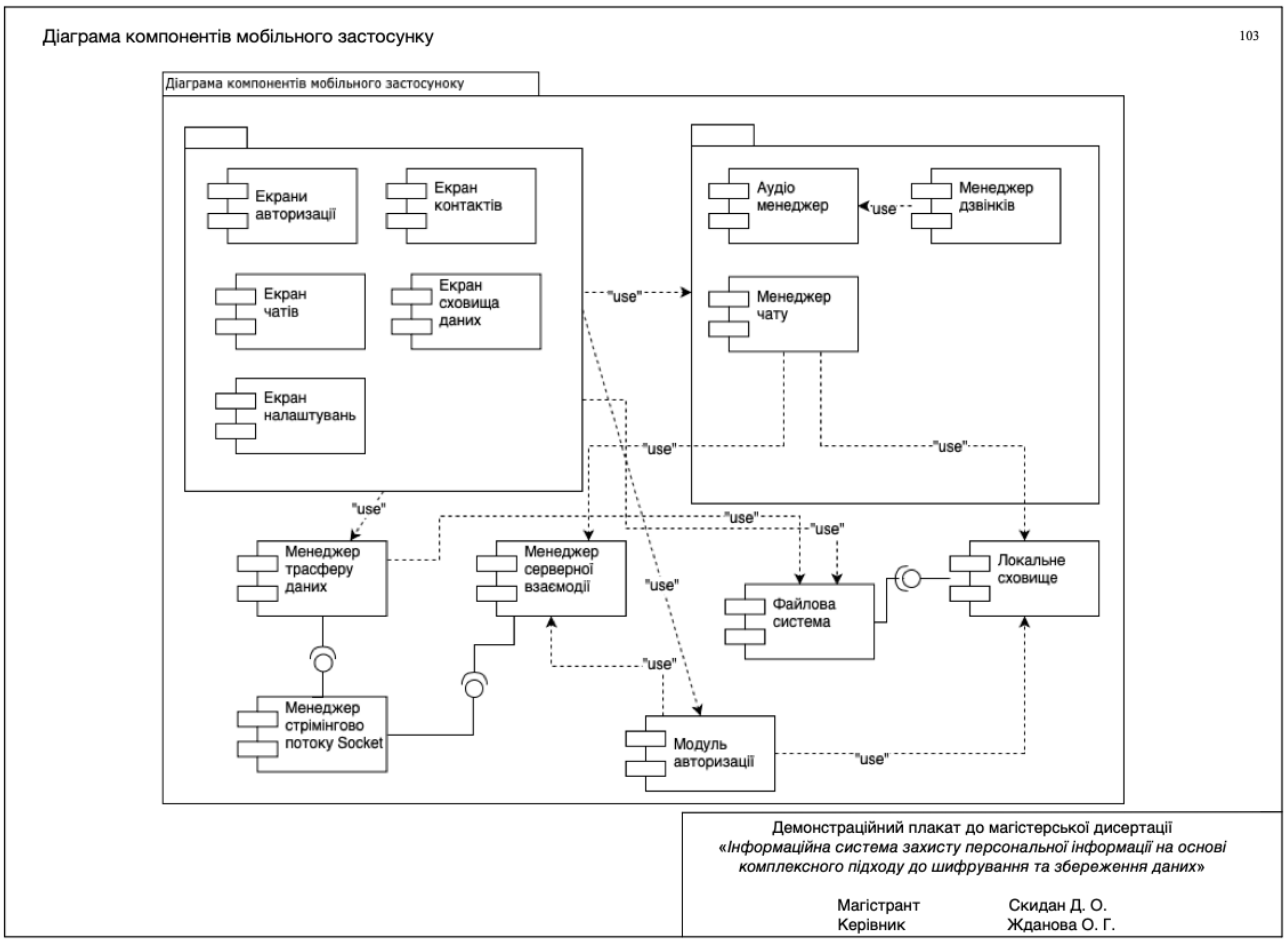
База даних мобільного застосунку



Демонстраційний плакат до магістерської дисертації
«*Інформаційна система захисту персональної інформації на основі комплексного підходу до шифрування та збереження даних*»

Магістрант Скидан Д. О.
Керівник Жданова О. Г.

Діаграма компонентів мобільного застосунку



Діаграма розгортання системи